

# High Order Cell-Centered Lagrangian-Type Finite Volume Schemes with Time-Accurate Local Time Stepping on Unstructured Triangular Meshes

Walter Boscheri<sup>a</sup>, Michael Dumbser<sup>\*a</sup>, Olindo Zanotti<sup>a</sup>

<sup>a</sup>Laboratory of Applied Mathematics, Department of Civil, Environmental and Mechanical Engineering  
University of Trento, Via Mesiano 77, I-38123 Trento, Italy

---

## Abstract

We present a novel cell-centered direct Arbitrary-Lagrangian-Eulerian (ALE) finite volume scheme on unstructured triangular meshes that is high order accurate in space and time and that also allows for *time-accurate local time stepping* (LTS). It extends our previous investigations on high order Lagrangian finite volume schemes with LTS carried out in [36] in one space dimension. The new scheme uses the following basic ingredients: a high order WENO reconstruction in space on unstructured meshes, an element-local high-order accurate space-time Galerkin predictor that performs the time evolution of the reconstructed polynomials within each element, the computation of numerical ALE fluxes at the moving element interfaces through approximate Riemann solvers, and a one-step finite volume scheme for the time update which is directly based on the integral form of the conservation equations in space-time. The inclusion of the LTS algorithm requires a number of crucial extensions, such as a proper scheduling criterion for the time update of each element and for each node; a virtual projection of the elements contained in the reconstruction stencils of the element that has to perform the WENO reconstruction; and the proper computation of the fluxes through the space-time boundary surfaces that will inevitably contain hanging nodes in time due to the LTS algorithm.

We have validated our new unstructured Lagrangian LTS approach over a wide sample of test cases solving the Euler equations of compressible gasdynamics in two space dimensions, including shock tube problems, cylindrical explosion problems, as well as specific tests typically adopted in Lagrangian calculations, such as the Kidder and the Saltzman problem. When compared to the traditional global time stepping (GTS) method, the newly proposed LTS algorithm allows to reduce the number of element updates in a given simulation by a factor that may depend on the complexity of the dynamics, but which can be as large as  $\sim 4.7$ .

**Keywords:** Arbitrary-Lagrangian-Eulerian (ALE), high order Lagrangian ADER-WENO schemes, moving unstructured meshes, time-accurate local time stepping (LTS), hyperbolic conservation laws, Euler equations of compressible gas dynamics

---

## 1. Introduction

In the last few years there has been a renewed interest in the development of novel accurate and robust cell-centered Lagrangian finite volume schemes for hydrodynamics. Since in a Lagrangian method the computational mesh moves with the local fluid velocity, such schemes are regarded as the first choice in all problems presenting moving material interfaces appearing in compressible multi-phase and multi-material flows, for instance, the numerical simulation of inertial confinement fusion (ICF). The vast majority of modern Lagrangian schemes adopts a *cell-centered* finite-volume approach, see for example [15, 23, 78, 68, 72, 73, 74, 71], where all flow variables are defined as cell-averaged quantities inside a control volume. However, also *staggered* Lagrangian schemes are possible, see e.g. [67], where the velocity is defined at the cell interfaces, while the other flow variables are still defined at the cell centers.

---

<sup>\*</sup>Corresponding author

Email addresses: `walter.boscheri@unitn.it` (Walter Boscheri), `michael.dumbser@unitn.it` (Michael Dumbser<sup>\*</sup>), `olindo.zanotti@unitn.it` (Olindo Zanotti)

In [75, 15] Godunov-type finite volume schemes have been presented for Lagrangian hydrodynamics, while in [25, 26] the governing equations have been coupled with the equations for the evolution of the geometry and the resulting weakly hyperbolic system has been solved using a node-based finite volume solver. Unstructured multidimensional meshes have been considered by Maire in [69, 71, 70], who developed up to second order accurate cell-centered Lagrangian schemes where the time derivatives of the fluxes have been computed with a node-centered solver. This approach may be regarded as a multi-dimensional Lagrangian extension of the Generalized Riemann problem methodology used for example in the ADER approach of Titarev and Toro [83, 85] in the Eulerian context. Arbitrary-Lagrangian-Eulerian (ALE) methods based on remeshing and remapping have also been investigated very recently for single and multi-material flows in [49, 90, 13, 79].

In [21, 64] Cheng and Shu presented the first better than second order accurate Lagrangian schemes for hydrodynamics on structured meshes, where the use of a high order Essentially Non-Oscillatory (ENO) reconstruction operator yielded high order of accuracy in space, while high order of accuracy in time was guaranteed using either a Runge-Kutta or a Lax-Wendroff-type time stepping. Arbitrary high order accurate cell-centered Lagrangian-type finite volume schemes for conservative and non-conservative hyperbolic PDE on moving unstructured triangular and tetrahedral meshes have been considered for the first time by Boscheri et al. in a very recent series of papers [10, 38, 12, 9, 11]. A new class of *meshless* Lagrangian particle methods based on a high order accurate moving least-squares WENO reconstruction has been forwarded in [2].

High order accurate Lagrangian algorithms using the classical continuous finite element method (FEM) can be found, for example, in the work of Scovazzi et al. [77, 80] and Dobrev et al. [28, 29, 30], while Lagrangian discontinuous Galerkin finite elements have been recently proposed by Vilar et al. and Yu et al. in [52, 50, 51, 63]. Arbitrary-Lagrangian-Eulerian DG schemes have been developed and applied, for example, in [20, 46].

Almost all of the above mentioned algorithms use an explicit *global* time stepping scheme in which the timestep is computed under a classical *global* CFL stability condition, so that the timestep is essentially determined by the smallest control volume appearing in the mesh. In Lagrangian hydrodynamics, where the mesh follows as closely as possible the local fluid motion, very severe deformations and distortions may occur in the computational cells, especially at shocks and shear waves. As a consequence, the computational efficiency of the algorithm drastically decreases, because the smallest timestep imposed by the most deformed control volumes dictates the timestep for the entire computational grid, including those elements which are much bigger or which lie in a zone where the fluid is moving uniformly. In the Eulerian framework such a problem can be partially avoided controlling the mesh quality *a priori* and designing a high quality mesh once in a pre-processing step, since the grid will not change anymore during the simulation. Of course, the CFL condition can be circumvented by using implicit or semi-implicit schemes, see for example [17, 19, 18, 31, 32, 33], but this approach does not yet seem to be very popular in the context of cell-centered Lagrangian-type finite volume methods. An alternative to overcome the global CFL condition consists in the development of numerical schemes that allow for time-accurate *local* time stepping (LTS), where each element has to obey only a less restrictive *local* CFL stability condition, hence using its own optimal local timestep. Therefore, many efforts have been devoted to the construction of high order accurate Eulerian schemes with time-accurate LTS, developing either discontinuous Galerkin finite element methods [48, 42, 82, 65, 53, 62, 47] or high order accurate finite volume schemes with LTS [8, 7, 89, 16, 4, 3, 14, 47, 45, 39]. The finite volume schemes with LTS adopt mainly classical adaptive mesh refinement (AMR) techniques in space and time or block-clustered local time stepping algorithms. In [55, 54] also high order accurate Runge-Kutta time integrators with local time stepping (so-called multi-rate integrators) can be found. To the knowledge of the authors, the first high order accurate *Lagrangian* algorithm with *time accurate local time stepping* on moving grids has been proposed very recently in [36], where the equations of hydrodynamics and of classical magnetohydrodynamics (MHD) have been solved in one spatial dimension. In the present paper we extend the algorithm presented in [36] to *moving unstructured triangular meshes*.

The rest of the paper is structured as follows: in Section 2 the numerical scheme is described, including the details of the local time stepping algorithm on moving unstructured meshes, while numerical convergence studies as well as some classical numerical test problems for hydrodynamics are presented in Section 3. We conclude the paper giving an outlook to future research and developments in Section 4.

## 2. Numerical method

### 2.1. Formulation of the equations and basic set-up

In this article we consider nonlinear hyperbolic conservation laws, cast in the form

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) = \mathbf{S}(\mathbf{Q}), \quad \mathbf{x} = (x, y) \in \Omega(t) \subset \mathbb{R}^2, \quad t \in \mathbb{R}_0^+, \quad \mathbf{Q} \in \Omega_{\mathbf{Q}} \subset \mathbb{R}^v, \quad (1)$$

where  $\mathbf{Q} = (q_1, q_2, \dots, q_v)$  is the vector of conserved variables defined in the space of the admissible states  $\Omega_{\mathbf{Q}} \subset \mathbb{R}^v$ ,  $\mathbf{F}(\mathbf{Q}) = (\mathbf{f}(\mathbf{Q}), \mathbf{g}(\mathbf{Q}))$  denotes the nonlinear flux tensor and  $\mathbf{S}(\mathbf{Q})$  represents a nonlinear algebraic source term which is not allowed to be stiff. The system of equations (1) is defined in two space dimensions, hence addressing the space coordinate vector and the time with  $\mathbf{x} = (x, y)$  and  $t$ , respectively. The two-dimensional computational domain  $\Omega(t)$  is time-dependent since in the Lagrangian framework the mesh is moving, hence changing its configuration at each time step. The domain is discretized using a total number of  $N_E$  conforming triangles  $T_i^n$ , therefore at a general time  $t^n$  the *current triangulation*  $\mathcal{T}_{\Omega}^n$  of the domain  $\Omega(t^n) = \Omega^n$  is given by the union of all elements, i.e.

$$\mathcal{T}_{\Omega}^n = \bigcup_{i=1}^{N_E} T_i^n. \quad (2)$$

Within the Lagrangian LTS algorithm that is going to be presented in this paper each element moves in time independently from the others and following its own local timestep, hence the triangulation  $\mathcal{T}_{\Omega}^n$  will in general *never* be assembled at a common time level. In the LTS framework hanging nodes in time are naturally produced and one is in general not able to define the configuration of the computational mesh at a certain time level  $t^n$ , unless we force the computation to reach the same time  $\bar{t}$ , which could be typically the case either at the final time or at an intermediate output time. For this reason in the rest of the paper each time level  $t^n$  will be addressed also with the element number it refers to, i.e.  $t_i^n$ , with the subscript  $i$  denoting the number of the physical triangle  $T_i$ .

As suggested in [10], we adopt a spatial reference system  $\xi - \eta$  defined by the coordinate vector  $\xi = (\xi, \eta)$  where the unit reference triangle  $T_e$  is composed of the nodes  $\xi_1^e = (\xi_1^e, \eta_1^e) = (0, 0)$ ,  $\xi_2^e = (\xi_2^e, \eta_2^e) = (1, 0)$  and  $\xi_3^e = (\xi_3^e, \eta_3^e) = (0, 1)$ . The physical element  $T_i^n$  defined in the physical system  $x - y$  is mapped to the reference element  $T_e$  using the transformation

$$\mathbf{x} = \mathbf{x}(\xi, t^n) = \mathbf{X}_{1,i}^n + (\mathbf{X}_{2,i}^n - \mathbf{X}_{1,i}^n)\xi + (\mathbf{X}_{3,i}^n - \mathbf{X}_{1,i}^n)\eta, \quad (3)$$

where  $\mathbf{X}_{k,i}^n = (X_{k,i}^n, Y_{k,i}^n)$  represents the vector of physical coordinates of the  $k$ -th vertex of triangle  $T_i^n$  at time  $t_i^n$ . In the Lagrangian framework the use of the reference system, which does not change in time, is much more convenient rather than carrying on the computation in the physical system, where elements are moving and deforming in time.

As usual for cell-centered finite volume schemes, data are represented and evolved in time within each control volume as piecewise constant cell averages

$$\mathbf{Q}_i^n = \frac{1}{|T_i^n|} \int_{T_i^n} \mathbf{Q}(\mathbf{x}, t_i^n) dV, \quad (4)$$

where the volume of element  $T_i^n$  is denoted by  $|T_i^n|$  at the current element time  $t_i^n$ . In the time-accurate LTS algorithm a cell  $T_i^n$  is allowed to evolve the solution in time only if the so-called *update criterion* [41, 65, 36] is satisfied, namely if

$$\max_{j \in \mathcal{N}_i} (t_j^n) \leq (t_i^n + \Delta t_i^n) \leq \min_{j \in \mathcal{N}_i} (t_j^n + \Delta t_j^n), \quad (5)$$

where  $\mathcal{N}_i$  denotes the *Neumann neighborhood* of element  $T_i$ , i.e. the three direct side neighbors  $T_j$  of the cell, while  $t_i^n$  and  $\Delta t_i^n$  represent the current local time and the local timestep of triangle  $T_i$ , respectively. Hence,  $(t_i^n + \Delta t_i^n)$  is the future time of element  $T_i$  and to make notation easier it will be addressed with  $t_i^{n+1}$ .

There are two important issues that need to be clarified:

1. In order to develop a numerical scheme that evolves the cell averages (4) with high order of accuracy in space and in time in one single step, two strategies are followed. For the accuracy in space we implement a suitable Weighted Essentially Non-Oscillatory (WENO) reconstruction technique that is able to deal with LTS and which is presented in detail in the next Section 2.2, while for the accuracy in time we use an element-local space-time Galerkin predictor approach, as illustrated in Section 2.3.

2. In a time-accurate LTS finite volume scheme, each element  $T_i^n$  evolves the solution  $\mathbf{Q}_i^n$  in time with a local timestep  $\Delta t_i^n$  that is computed according to a local CFL stability condition. As a result, the WENO reconstruction will be carried out *locally*, i.e. considering only the element  $T_i^n$  which is currently updating the solution to its new time level  $t_i^{n+1}$ , as well as an appropriate neighborhood of  $T_i^n$  that is necessary to carry out the reconstruction, the so-called reconstruction stencil  $\mathcal{S}_i^W$ . Since the neighbor elements of  $T_i$  in general have a different local time, the reconstruction needs to get time-accurate *virtual* cell averages from the neighbor cells as input. These virtual cell averages are readily available from the local space-time Galerkin predictor solution inside the neighbors.

## 2.2. High order WENO reconstruction for local time stepping

In order to obtain high order of accuracy in space a nonlinear WENO reconstruction algorithm is used. As done in [10, 38, 12, 9, 11] we adopt the *polynomial* formulation presented in [41, 40, 84, 88], instead of the original *pointwise* approach proposed by Shu et al. in [58, 5, 57, 92]. Other high order accurate reconstruction algorithms on unstructured meshes can be found, e.g. in [1, 22, 27, 66]. While all the details of high order WENO reconstruction are contained in the above-mentioned references, we present here only a brief summary of the main features of the scheme, highlighting the modifications that are necessary to handle a time accurate local time stepping formulation.

The reconstructed solution  $\mathbf{w}_h(\mathbf{x}, t_i^n)$  is given in terms of piecewise polynomials of degree  $M$  and is computed *locally* for each control volume  $T_i^n$ . First, one has to construct a set of reconstruction stencils  $\mathcal{S}_i^s$  relative to the element  $T_i$ , namely

$$\mathcal{S}_i^s = \bigcup_{j=1}^{n_e} T_{m(j)}, \quad (6)$$

where  $1 \leq j \leq n_e$  denotes a local index which counts the elements belonging to the stencil, while  $m(j)$  maps the local counter  $j$  to the global element number used in the triangulation (2). As explained in [6, 76, 60, 40], in two space dimensions on unstructured meshes one has to take a total number of elements  $n_e$  for each stencil that is bigger than the smallest number  $\mathcal{M} = (M+1)(M+2)/2$  needed to reach the formal order of accuracy  $M+1$ , hence we typically set  $n_e = 2\mathcal{M}$ . Furthermore, according to [60, 40], we need a total number of stencils  $s = 7$  in order to perform the polynomial WENO reconstruction, namely one central stencil  $s = 1$ , three primary sector stencils  $s \in \{2, 3, 4\}$  and three reverse sector stencils  $s \in \{5, 6, 7\}$ . As a consequence, the update criterion (5) must be extended to the *total* WENO stencil  $\mathcal{S}_i^W$  given by

$$\mathcal{S}_i^W = \bigcup_{s=1}^7 \mathcal{S}_i^s, \quad (7)$$

hence obtaining

$$\max(t_j^n) \leq t_i^{n+1} \leq \min(t_j^{n+1}), \quad \forall T_j \in \mathcal{S}_i^W. \quad (8)$$

In order to guarantee that at least one element in the entire mesh satisfies condition (8), the total stencils  $\mathcal{S}_i^W$  need to be constructed in such a way that they are *symmetric*, i.e. each element  $T_j \in \mathcal{S}_i^W$  inside the stencil of  $T_i$  *must* contain in its own WENO stencil  $\mathcal{S}_j^W$  the element  $T_i$ . In other words, if  $T_j \in \mathcal{S}_i^W$  then  $T_i \in \mathcal{S}_j^W$ . It is always possible to construct such symmetric stencils by adding elements to the stencils until the condition of symmetry is satisfied for all elements.

For the sake of clarity we give a simple example of what could happen if we take *non-symmetric* stencils. Let element  $T_j$  be *not* contained in the stencil of  $T_i$  and let  $T_i$  belong to the stencil  $\mathcal{S}_j^W$  of element  $T_j$ . Let furthermore the current time level of  $T_i$  and  $T_j$  be  $t_i^n$  and  $t_j^n$ , respectively, with the corresponding future times  $t_i^{n+1}$  and  $t_j^{n+1}$ . Without loss of generality we assume  $t_i^n = t_j^n$ , while the future time levels are chosen such that  $t_i^{n+1} > t_j^{n+1}$ . If the update criterion on the *non-symmetric* stencil  $\mathcal{S}_i^W$  is supposed to be satisfied, then element  $T_i$  is allowed to update the numerical solution to its future time, which will subsequently become the *current* time of  $T_i$ , i.e.  $t_i^n \rightarrow t_i^{n+1}$ . The resulting situation will lead to a *dead lock* in the algorithm, where element  $T_j$  will never obey condition (8) since  $t_j^{n+1} < t_i^n$ . A simple solution is to always build a *symmetric* stencil. In this case element  $T_j$  performs the update *first* and does not prevent element  $T_i$  from updating its solution. The drawback of this approach is that slightly larger stencils are required.

Due to (8), the current time  $t_j^n$  of the neighbor elements belonging to the WENO stencil  $\mathcal{S}_i^W$  must be lower than the current time level  $t_i^n$  of the triangle  $T_i$  for which the reconstruction has to be performed. Moreover, in Lagrangian

algorithms the mesh is moving in time, therefore the local WENO reconstruction is carried out on a *virtual geometry* with *virtual cell averages*, as suggested in [36]. These virtual cell averages, which are needed for the reconstruction, are obtained from the local space-time predictor solution  $\mathbf{q}_h(\mathbf{x}, t_i^n)$  inside the neighbor elements  $T_j$  using a simple integral *projection* (averaging). The way how this predictor solution is computed will be described in the next Section 2.3. A similar projection is used also for the virtual geometry of the elements inside the total WENO stencil, where all elements  $T_j^n \in \mathcal{S}_i^W$  are moved *virtually* until time  $t_i^n$  is reached. We emphasize that the projection of the stencil geometry and of the cell averages is done only virtually, just for the purpose of reconstruction, because the real mesh motion and the real conservative update of the cell averages will be performed individually by each element at its scheduled time according to the update criterion (8). The geometry of each stencil element  $T_j^n$ , i.e. the vertex coordinates, are projected and also all the other geometric quantities used for the computation, e.g. normal vectors, volumes, side lengths, *etc.*. For the sake of clarity, the projected quantities will be denoted by a tilde symbol in the following, hence

$$\tilde{\mathbf{X}}_{k,j}^{n+1} = \mathbf{X}_{k,j}^n + (t_i^n - t_j^n) \bar{\mathbf{V}}_{k,j}^n, \quad \forall T_j^n \in \mathcal{S}_i^W, \quad k = 1, 2, 3 \quad (9)$$

and

$$\tilde{\mathbf{Q}}_j^n = \begin{cases} \mathbf{Q}_i^n, & \text{if } j = i, \\ \frac{1}{|\tilde{T}_j^n|} \int_{\tilde{T}_j^n} \mathbf{q}_h(\mathbf{x}, t_i^n) dV, & \text{if } j \neq i, \end{cases} \quad \forall T_j^n \in \mathcal{S}_i^W. \quad (10)$$

In (9) the time-averaged node velocity  $\bar{\mathbf{V}}_{k,j}^n$  is computed according to the node solver algorithm, see also [10, 11, 12, 9], which will be briefly described in Section 2.4, while in (10) the virtual cell averages  $\tilde{\mathbf{Q}}_j^n$  of the neighbor elements are given as the spatial integral of the predicted solution at time  $t_i^n$  over the virtual control volumes  $\tilde{T}_j^n$ .

Once the virtual geometry and the virtual cell averages have been computed for the entire stencil  $\mathcal{S}_i^W$ , we are in the position to carry out the *local* high order WENO reconstruction procedure. To obtain the reconstruction polynomial  $\mathbf{w}_h(\mathbf{x}, t_i^n)$ , integral conservation of the projected cell averages  $\tilde{\mathbf{Q}}_j^n$  in each reconstruction stencil  $\mathcal{S}_i^S$  is required, i.e.

$$\frac{1}{|\tilde{T}_j^n|} \int_{\tilde{T}_j^n} \mathbf{w}_h^s(\mathbf{x}, t_i^n) dV = \frac{1}{|\tilde{T}_j^n|} \int_{\tilde{T}_j^n} \psi_l(\xi, \eta) \hat{\mathbf{w}}_{l,i}^{n,s} = \tilde{\mathbf{Q}}_j^n, \quad \forall T_j^n \in \mathcal{S}_i^S, \quad (11)$$

where the integrals are evaluated using Gaussian quadrature formulae of suitable order (see [81] for details). For simplicity, in the above equation, as well as in the rest of the paper, we have adopted the Einstein summation convention over repeated indices. The reconstruction polynomial on each stencil is expressed in terms of a set of orthogonal spatial basis functions  $\psi_l(\xi, \eta)$  on the reference element, see [34, 59, 24], and  $\mathcal{M}$  unknown degrees of freedom  $\hat{\mathbf{w}}_{l,i}^{n,s}$ . Since each stencil contains a total number of elements  $n_e > \mathcal{M}$ , system (11) results in an overdetermined linear algebraic system that is solved by a constrained least-squares technique [40]. In the Lagrangian framework the geometry evolves in time. Hence, the reconstruction matrix, which is given by the multidimensional integrals in (11), continuously changes in time. As a consequence, the system (11) must be solved whenever element  $T_i$  performs its WENO reconstruction. To maintain the scheme as simple as possible and reasonably cost efficient, the stencil topology is fixed once and for all in a preprocessing stage and is not dynamically recomputed.

In order to avoid spurious oscillations at discontinuities, the reconstruction operator must be nonlinear. Therefore the polynomials defined on each stencil are combined with each other and weighted in a nonlinear way, where the non-linearity is introduced in the WENO weights  $\omega_s$

$$\tilde{\omega}_s = \frac{\lambda_s}{(\sigma_s + \epsilon)^r}, \quad \omega_s = \frac{\tilde{\omega}_s}{\sum_q \tilde{\omega}_q}, \quad (12)$$

through the oscillation indicators  $\sigma_s$ , which are computed according to [58, 41, 40] as

$$\sigma_s = \sum_{lm} \hat{\mathbf{w}}_{l,i}^{n,s} \hat{\mathbf{w}}_{m,i}^{n,s}, \quad (13)$$

with

$$\Sigma_{lm} = \sum_{\alpha+\beta \leq M} \int_{T_e} \frac{\partial^{\alpha+\beta} \psi_l(\xi, \eta)}{\partial \xi^\alpha \partial \eta^\beta} \cdot \frac{\partial^{\alpha+\beta} \psi_m(\xi, \eta)}{\partial \xi^\alpha \partial \eta^\beta} d\xi d\eta. \quad (14)$$



As done in [40, 41], we set  $\epsilon = 10^{-14}$ ,  $r = 8$ ,  $\lambda_s = 1$  for the one-sided stencils ( $s > 1$ ) and  $\lambda_1 = 10^5$  for the central stencil. The final nonlinear WENO reconstruction polynomial and its coefficients are then given by

$$\mathbf{w}_h(x, y, t_i^n) = \psi_l(\xi, \eta) \hat{\mathbf{w}}_{l,i}^n, \quad \text{with} \quad \hat{\mathbf{w}}_{l,i}^n = \sum_{s=1}^7 \omega_s \hat{\mathbf{w}}_{l,i}^{n,s}. \quad (15)$$

### 2.3. Local space-time Galerkin predictor on moving triangles

In order to achieve high order of accuracy in time we use the local space-time continuous Galerkin method, where the reconstructed polynomial  $\mathbf{w}_h$  obtained at the current element time  $t_i^n$  are *evolved* locally within element  $T_i(t)$  until the future time  $t_i^{n+1}$ . This method was first introduced for the Eulerian framework in [37] and then extended to moving meshes in [44, 10, 38, 11]. In all the above-mentioned references the space-time continuous Galerkin procedure has been proposed *locally*, i.e. the high order evolution of the reconstructed polynomial has always been carried out within each control volume and considering separately all the elements of the entire mesh. As a consequence, such a procedure automatically fits the construction of a time-accurate local time stepping algorithm.

As previously done for the WENO reconstruction, we use again the spatial reference system  $\xi - \eta$ , where now the relative time  $\tau$  is also considered. Therefore the physical element can be mapped to the reference space-time element  $T_E \times [0, 1]$  using the local space transformation (3) and the following mapping in time:

$$t = t_i^n + \tau \Delta t_i^n, \quad \tau = \frac{t - t_i^n}{\Delta t_i^n}. \quad (16)$$

The spatial coordinate vector in physical and reference coordinates are given by  $\mathbf{x} = (x, y)$  and  $\boldsymbol{\xi} = (\xi, \eta)$ , respectively, while  $\tilde{\mathbf{x}} = (x, y, t)$  and  $\tilde{\boldsymbol{\xi}} = (\xi, \eta, \tau)$  are the corresponding space-time coordinate vectors. According to [37], we adopt a *weak integral formulation* of the governing PDE (1), which is rewritten in the space-time reference system using the relations (3)-(16):

$$\frac{\partial \mathbf{Q}}{\partial \tau} \tau_t + \frac{\partial \mathbf{Q}}{\partial \xi} \xi_t + \frac{\partial \mathbf{Q}}{\partial \eta} \eta_t + \frac{\partial \mathbf{f}}{\partial \tau} \tau_x + \frac{\partial \mathbf{f}}{\partial \xi} \xi_x + \frac{\partial \mathbf{f}}{\partial \eta} \eta_x + \frac{\partial \mathbf{g}}{\partial \tau} \tau_y + \frac{\partial \mathbf{g}}{\partial \xi} \xi_y + \frac{\partial \mathbf{g}}{\partial \eta} \eta_y = \mathbf{S}(\mathbf{Q}). \quad (17)$$

The Jacobian of the spatial and temporal transformation and its inverse read

$$J_{st} = \frac{\partial \tilde{\mathbf{x}}}{\partial \tilde{\boldsymbol{\xi}}} = \begin{pmatrix} x_\xi & x_\eta & x_\tau \\ y_\xi & y_\eta & y_\tau \\ 0 & 0 & \Delta t \end{pmatrix}, \quad J_{st}^{-1} = \frac{\partial \tilde{\boldsymbol{\xi}}}{\partial \tilde{\mathbf{x}}} = \begin{pmatrix} \xi_x & \xi_y & \xi_t \\ \eta_x & \eta_y & \eta_t \\ 0 & 0 & \frac{1}{\Delta t} \end{pmatrix}, \quad (18)$$

where we used the properties  $\tau_x = \tau_y = 0$  and  $\tau_t = \frac{1}{\Delta t}$ , according to the definition (16). We rely on the inverse of the Jacobian matrix for reducing Eqn. (17) to

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \Delta t \left( \frac{\partial \mathbf{Q}}{\partial \xi} \xi_t + \frac{\partial \mathbf{Q}}{\partial \eta} \eta_t + \frac{\partial \mathbf{f}}{\partial \xi} \xi_x + \frac{\partial \mathbf{f}}{\partial \eta} \eta_x + \frac{\partial \mathbf{g}}{\partial \xi} \xi_y + \frac{\partial \mathbf{g}}{\partial \eta} \eta_y \right) = \Delta t \mathbf{S}(\mathbf{Q}), \quad (19)$$

which can be simply reformulated as

$$\mathbf{Q}_\tau = \Delta t \mathbf{P}, \quad (20)$$

with the aid of the term  $\mathbf{P}$  defined as

$$\mathbf{P} := \mathbf{S}(\mathbf{Q}) - \left( \frac{\partial \mathbf{Q}}{\partial \xi} \xi_t + \frac{\partial \mathbf{Q}}{\partial \eta} \eta_t + \frac{\partial \mathbf{f}}{\partial \xi} \xi_x + \frac{\partial \mathbf{f}}{\partial \eta} \eta_x + \frac{\partial \mathbf{g}}{\partial \xi} \xi_y + \frac{\partial \mathbf{g}}{\partial \eta} \eta_y \right). \quad (21)$$

As done in [37], the solution vector  $\mathbf{Q}$ , the flux tensor  $\mathbf{F}$ , the source term  $\mathbf{S}$  as well as the term  $\mathbf{P}$  are discretized using a nodal finite element approach. The discrete solutions are denoted by  $\mathbf{q}_h$ ,  $\mathbf{F}_h$ ,  $\mathbf{S}_h$  and  $\mathbf{P}_h$ , respectively, and are given by

$$\begin{aligned} \mathbf{q}_h &= \mathbf{q}_h(\xi, \eta, \tau) = \theta_l(\xi, \eta, \tau) \hat{\mathbf{q}}_{l,i}, & \mathbf{S}_h &= \mathbf{S}_h(\xi, \eta, \tau) = \theta_l(\xi, \eta, \tau) \hat{\mathbf{S}}_{l,i}, \\ \mathbf{F}_h &= \mathbf{F}_h(\xi, \eta, \tau) = \theta_l(\xi, \eta, \tau) \hat{\mathbf{F}}_{l,i}, & \mathbf{P}_h &= \mathbf{P}_h(\xi, \eta, \tau) = \theta_l(\xi, \eta, \tau) \hat{\mathbf{P}}_{l,i}. \end{aligned} \quad (22)$$

Here,  $\theta_l = \theta_l(\tilde{\xi}) = \theta_l(\xi, \eta, \tau)$  are a set of space-time nodal basis functions defined by the Lagrange interpolation polynomials passing through a set of space-time nodes  $\tilde{\xi}_m = (\xi_m, \eta_m, \tau_m)$ , see [37] for details. The same approximation also applies to the mapping from the physical space-time coordinate vector  $\tilde{\mathbf{x}}$  to the reference space-time coordinate vector  $\tilde{\xi}$ , hence

$$\mathbf{x}(\xi, \eta, \tau) = \theta_l(\xi, \eta, \tau) \widehat{\mathbf{x}}_{l,i}, \quad t(\xi, \eta, \tau) = \theta_l(\xi, \eta, \tau) \widehat{t}_l, \quad (23)$$

where the use of the *same* basis functions  $\theta_l$  is allowed by the adoption of an *isoparametric* approach.  $\widehat{\mathbf{x}}_{l,i} = (\widehat{x}_{l,i}, \widehat{y}_{l,i})$  are the degrees of freedom of the vector of physical coordinates in space, that are partially unknown, while  $\widehat{t}_l$  denote the *known* degrees of freedom of the physical time at each space-time node  $\tilde{\mathbf{x}}_{l,i} = (\widehat{x}_{l,i}, \widehat{y}_{l,i}, \widehat{t}_l)$  according to (16).

In order to obtain the weak formulation of the governing PDE (1), we first multiply (19) with a test function which is given by the same space-time basis functions  $\theta_k(\xi, \eta, \tau)$  and then we integrate it over the unit reference space-time element  $T_e \times [0, 1]$ , i.e.

$$\left\langle \theta_k, \frac{\partial \theta_l}{\partial \tau} \right\rangle \widehat{\mathbf{q}}_{l,i} = \Delta t \langle \theta_k, \theta_l \rangle \widehat{\mathbf{P}}_{l,i}, \quad (24)$$

where the approximations given by (22) have been used as well as the following integral operator

$$\langle f, g \rangle = \int_0^1 \int_{T_e} f(\xi, \eta, \tau) g(\xi, \eta, \tau) d\xi d\eta d\tau, \quad (25)$$

which denotes the scalar product of two functions  $f$  and  $g$  over the space-time reference element  $T_e \times [0, 1]$ . Moreover the universal matrices

$$\mathbf{K}_\tau = \left\langle \theta_k, \frac{\partial \theta_l}{\partial \tau} \right\rangle \quad \text{and} \quad \mathbf{M} = \langle \theta_k, \theta_l \rangle \quad (26)$$

proposed in [10, 38] are used to write expression (24) in a more compact matrix notation, namely

$$\mathbf{K}_\tau \widehat{\mathbf{q}}_{l,i} = \Delta t \mathbf{M} \widehat{\mathbf{P}}_{l,i}. \quad (27)$$

Let  $\widehat{\mathbf{q}}_{l,i}^0$  be the part of the degrees of freedom of vector  $\widehat{\mathbf{q}}_{l,i}$  that are known from the initial condition  $\mathbf{w}_h$  by setting the corresponding degrees of freedom to the known values (see [37] for details) and let  $\widehat{\mathbf{q}}_{l,i}^1$  represent the unknown degrees of freedom for  $\tau > 0$ . Since  $\widehat{\mathbf{q}}_{l,i}^0$  are known, they can be moved onto the right-hand side of (27), hence obtaining the following nonlinear algebraic equation system (20), which can be solved by an iterative procedure, i.e.

$$\mathbf{K}_\tau \widehat{\mathbf{q}}_{l,i}^{r+1} = \Delta t \mathbf{M} \widehat{\mathbf{P}}_{l,i}^r, \quad (28)$$

with the superscript  $r$  denoting the iteration number. The initial guess ( $r = 0$ ) can be simply given by the reconstruction polynomial  $\mathbf{w}_h$  at the initial time level, otherwise a more efficient initial condition based on a second order MUSCL-type scheme can be used (see [56]).

Due to the Lagrangian formulation, which implies mesh motion, we have also to consider the evolution of the vertex coordinates of the local space-time element. The motion is governed by the following ODE system

$$\frac{d\mathbf{x}}{dt} = \mathbf{V}(x, y, t), \quad (29)$$

with the local mesh velocity  $\mathbf{V} = \mathbf{V}(x, y, t) = (U, V)$  approximated again with a nodal approach as

$$\mathbf{V}_h = \mathbf{V}_h(\xi, \eta, \tau) = \theta_l(\xi, \eta, \tau) \widehat{\mathbf{V}}_{l,i}, \quad \widehat{\mathbf{V}}_{l,i} = \mathbf{V}(\tilde{\mathbf{x}}_{l,i}). \quad (30)$$

Our algorithm belongs to the family of the so-called Arbitrary-Lagrangian-Eulerian (ALE) schemes, hence we allow the mesh velocity to be potentially different from the local fluid velocity. In this way Eulerian algorithms are reproduced by setting the mesh velocity to zero, while almost pure Lagrangian methods can be obtained when the mesh velocity coincides with the local fluid velocity. As suggested in [44, 10] the system (29) can be conveniently solved for the unknown coordinate vector  $\widehat{\mathbf{x}}_{l,i}$  using the same local space-time Galerkin method:

$$\left\langle \theta_k, \frac{\partial \theta_l}{\partial \tau} \right\rangle \widehat{\mathbf{x}}_{l,i} = \Delta t \langle \theta_k, \theta_l \rangle \widehat{\mathbf{V}}_{l,i}, \quad (31)$$

which yields the iteration scheme

$$\mathbf{K}_\tau \widehat{\mathbf{x}}_{l,i}^{r+1} = \Delta t \mathbf{M} \widehat{\mathbf{V}}_{l,i}^r. \quad (32)$$

Since the physical triangle  $T_i^n$  at time  $t_i^n$  is known, the initial condition of the ODE system is simply given by the nodal degrees of freedom  $\widehat{\mathbf{x}}_l$  at relative time  $\tau = 0$ .

In practice, the ODE system (29) is solved at each iteration of the PDE solver (28) and the procedure is repeated until convergence is reached. At the end of the local space-time Galerkin procedure we obtain an *element-local predictor* for the numerical solution  $\mathbf{q}_h$ , for the fluxes  $\mathbf{F}_h = (\mathbf{f}_h, \mathbf{g}_h)$ , for the source term  $\mathbf{S}_h$  and also for the mesh velocity  $\mathbf{V}_h$ .

In a Lagrangian scheme with LTS we are dealing with hanging nodes in time and we generally do not have a matching in time of the geometry, as already explained before, but discontinuities in the geometry configuration are not admitted. In cell-centered Lagrangian schemes a unique node velocity is obtained by a so-called no-solver algorithm that takes as input all vertex-extrapolated states from the triangles in the Voronoi neighborhood surrounding the vertex. In the Lagrangian ADER-WENO schemes with global time stepping presented in [10, 38] we used a suitable node solver algorithm to update the mesh *globally*, since the future time was the same for all the elements. Here, in the context of LTS, we adopt again the node solver algorithm with the aim to fix a *unique* node velocity, but the vertex will be *physically* (and not virtually) moved only when an element of the Voronoi neighborhood of the vertex fulfills the update criterion (8). To handle this situation in practice, each node  $k$  is also equipped with a local node time  $t_k^n$ .

#### 2.4. Mesh motion with local time stepping

As explained at the end of Section 2.3, each node  $k$  of the computational mesh needs to be assigned a *uniquely* defined velocity vector. The Voronoi neighborhood  $\mathcal{V}_k$  of node  $k$  is composed by all those elements  $T_j$  which share the node  $k$ . The node  $k$  will be moved each time the update criterion (8) is satisfied by one element  $T_i \in \mathcal{V}_k$ . Therefore the future time to which node  $k$  moves will coincide with the future time  $t_i^{n+1}$  of that element  $T_i$ .

In [12] three different node solver algorithms have been presented and here we consider the node solver denoted as  $\mathcal{NS}_{cs}$ , which adopts the idea of Cheng and Shu [21, 64]. However, rather than taking a simple arithmetic average of the velocity, the node velocity  $\bar{\mathbf{V}}_k$  is computed as a *mass weighted* average velocity among the neighborhood  $\mathcal{V}_k$  of node  $k$ , i.e.

$$\bar{\mathbf{V}}_k = \frac{1}{\mu_k} \sum_{T_j \in \mathcal{V}_k} \mu_{k,j} \bar{\mathbf{V}}_{k,j}, \quad (33)$$

with

$$\mu_k = \sum_{T_j \in \mathcal{V}_k} \mu_{k,j}, \quad \mu_{k,j} = \rho_j^n |T_j^n|. \quad (34)$$

The local weights  $\mu_{k,j}$  are the masses of the elements  $T_j$ , obtained by multiplying the cell averages of the density  $\rho_j$  with the cell area  $|T_j|$  at the current neighbor time level  $t_j^n$ .

The mesh motion plays an important role in Lagrangian schemes, because it allows interfaces and shear waves to be precisely identified. For this reason an accurate computation of the node velocity represents a crucial step, and in our approach the local velocity contributions  $\bar{\mathbf{V}}_{k,j}$  are taken to be the time integrals of the high order vertex-extrapolated velocities at node  $k$ . We can use the space-time reference system  $\xi - \eta - \tau$  and the velocity approximation given by (30) to evaluate the time integral. Since each node  $k$  can be moved by any of the Voronoi neighbors  $T_j$ , the vertex time level of node  $k$  is not known *a priori* when an element  $T_i$  satisfies (8) and is ready to update the geometry. Therefore, it is much more convenient to define a *node time* variable  $t_k^n$ , that is independent of the time evolution of the elements and advances in time whenever the node is moved by any of its Voronoi neighbors  $T_j$ . As a result, the high order velocity integration for each element  $T_j \in \mathcal{V}_k$  must be done within the time interval  $\Delta t_k = [t_k^n, t_k^{n+1}]$ , that has to be *rescaled* to the corresponding reference time interval  $\Delta \tau_k = [\tau_{k,j}^0, \tau_{k,j}^1]$  as

$$\tau_{k,j}^0 = \frac{t_k^n - t_j^n}{\Delta t_j^n}, \quad \tau_{k,j}^1 = \frac{t_k^{n+1} - t_j^n}{\Delta t_j^n}, \quad \forall T_j \in \mathcal{V}_k, \quad (35)$$



where  $\Delta t_j^n$  is the local timestep of element  $T_j$ . Recall that  $t_k^{n+1} = t_i^{n+1}$ , if the node is moved by element  $T_i$  which is supposed to satisfy the update criterion. Finally the local velocity contributions  $\bar{\mathbf{V}}_{k,j}$  are given by

$$\bar{\mathbf{V}}_{k,j} = \left( \int_{\tau_{k,j}^0}^{\tau_{k,j}^1} \theta_l(\xi_{m(k)}^e, \eta_{m(k)}^e, \tau) d\tau \right) \widehat{\mathbf{V}}_{l,j}, \quad (36)$$

where  $m(k)$  is a mapping from the global node number  $k$  to the local node number in element  $T_j$ , while  $\xi_m^e$  and  $\eta_m^e$  represent the coordinates of the vertices of the reference triangle in space.  $\widehat{\mathbf{V}}_{l,j}$  are the space-time degrees of freedom which are *known* from the local space-time predictor solution  $\mathbf{q}_{h,j}$ . Each node  $k$  belonging to element  $T_i$  is finally moved to the new position  $\mathbf{X}_k^{n+1}$  with

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^n + \Delta t_k \bar{\mathbf{V}}_k. \quad (37)$$

## 2.5. Finite volume scheme

The vector of conserved variables  $\mathbf{Q}_i^n$  is evolved to the next time level  $t_i^{n+1}$  only when element  $T_i$  obeys the update criterion (8). As proposed in [10, 11] the governing PDE (1) can be rewritten in a more compact space-time divergence form, which reads

$$\tilde{\nabla} \cdot \tilde{\mathbf{F}} = \mathbf{S}(\mathbf{Q}), \quad (38)$$

with the space-time nabla operator and the tensor  $\tilde{\mathbf{F}}$  defined as

$$\tilde{\nabla} = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial t} \right)^T, \quad \tilde{\mathbf{F}} = (\mathbf{F}, \mathbf{Q}) = (\mathbf{f}, \mathbf{g}, \mathbf{Q}). \quad (39)$$

The conservation law (38) is then integrated in space and time over the space-time control volume  $C_i^n = T_i(t) \times [t_i^n, t_i^{n+1}]$  generated by the time evolution of element  $T_i$  and depicted in Figure 1, hence yielding

$$\int_{t_i^n}^{t_i^{n+1}} \int_{T_i(t)} \tilde{\nabla} \cdot \tilde{\mathbf{F}} d\mathbf{x} dt = \int_{t_i^n}^{t_i^{n+1}} \int_{T_i(t)} \mathbf{S} d\mathbf{x} dt, \quad (40)$$

which, after application of Gauss' theorem, reads

$$\int_{\partial C_i^n} \tilde{\mathbf{F}} \cdot \tilde{\mathbf{n}} dS = \int_{t_i^n}^{t_i^{n+1}} \int_{T_i(t)} \mathbf{S} d\mathbf{x} dt, \quad (41)$$

The vector  $\tilde{\mathbf{n}} = (\tilde{n}_x, \tilde{n}_y, \tilde{n}_t)$  is the outward pointing space-time unit normal vector defined on the space-time surface  $\partial C_i^n$ , which is composed of five space-time sub-surfaces, as shown in Figure 1: the first one  $\partial C_{bot}^n$  is given by the element configuration  $T_i^n$  at the current time level, while  $\partial C_{top}^n$  represents the control volume  $T_i^{n+1}$  evolved to the future time level. The remaining three lateral space-time sub-surfaces  $\partial C_{ij}^n$  are usually shared with the so-called *Neumann neighbors*  $N_i$  of  $T_i$ , i.e. with the direct side neighbors. As explained in [10, 38, 12] a set of bilinear basis functions are used to parametrize the lateral sub-surfaces, which are mapped onto a side-aligned local reference system  $(\chi, \tau)$ . The unit normal vector  $\tilde{\mathbf{n}}$  can be computed from the parametrization of the lateral sub-surfaces, while for  $\partial C_{bot}^n$  and  $\partial C_{top}^n$  it simply reads  $\tilde{\mathbf{n}} = (0, 0, -1)$  and  $\tilde{\mathbf{n}} = (0, 0, 1)$ , respectively.

In the time-accurate local time stepping (LTS) algorithm, when the element  $T_i$  is ready to update its numerical solution  $\mathbf{Q}_i^n$ , it might well be the case that the vertices of  $T_i$  have already been moved by another element  $T_j$  sharing one or more nodes with  $T_i$ . This situation generates hanging nodes in time, as shown in Figure 1, where vertex 1 has changed its position to  $1'$ . In order to design a suitable finite volume scheme on moving meshes with LTS, some parts of the flux integral appearing in (41) will be computed using a *memory variable*  $\mathbf{Q}_i^M$ , according to [36]. The memory variable contains all fluxes through the element space-time sub-surfaces  $\partial C_{ij}^n$  in the *past*, e.g. the fluxes through the

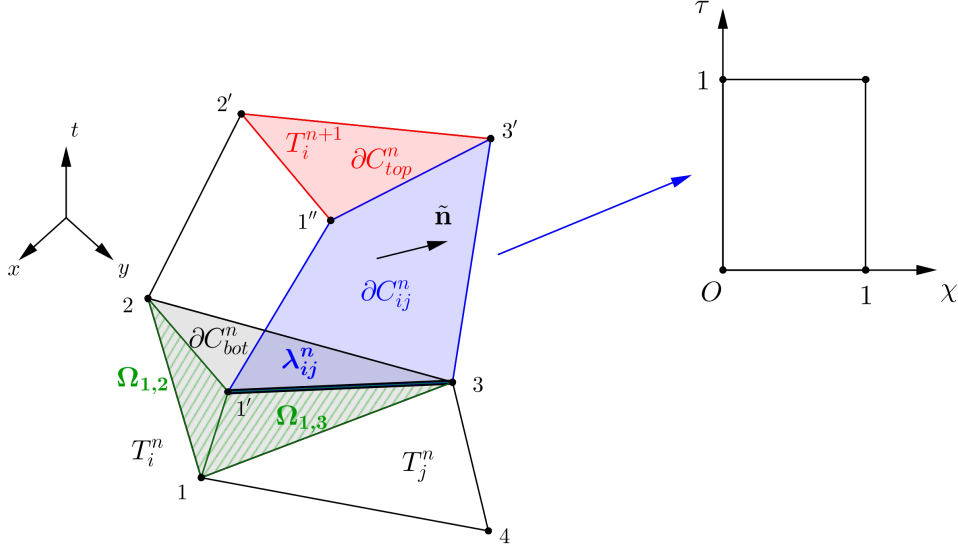


Figure 1: Space-time evolution of element  $T_i$  from time  $t_i^n$  (black triangle) to time  $t_i^{n+1}$  (red triangle). The triangular sub-surfaces  $\Omega_{1,2}$  and  $\Omega_{2,3}$  (already computed in the past by some Voronoi neighbors of the vertices of  $T_i$ ) are highlighted in green, while the trapezoidal space-time sub-surfaces  $\partial C_{ij}^n$  computed with the current element update are highlighted in blue.

space-time triangular surfaces  $\Omega_{1,2}$  and  $\Omega_{1,3}$  depicted in Figure 1. Therefore, from (41) the following high order ALE one-step finite volume scheme with LTS is obtained:

$$|T_i^{n+1}| \mathbf{Q}_i^{n+1} = |T_i^n| \mathbf{Q}_i^n - \sum_{T_j \in \mathcal{N}_i} \int_0^1 \int_0^1 |\partial C_{ij}^n| \tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} d\tau d\chi + \int_{t_i^n}^{t_i^{n+1}} \int_{T_i(t)} \mathbf{S}(\mathbf{q}_h) d\mathbf{x} dt + \mathbf{Q}_i^M, \quad (42)$$

with  $|T_i^n|$  and  $|T_i^{n+1}|$  representing the surface of triangle  $T_i$  at the current and at the future time level, i.e.  $t_i^n$  and  $t_i^{n+1}$ , and  $|\partial C_{ij}^n|$  denoting the determinant of the coordinate transformation of each lateral sub-surface  $\partial C_{ij}^n$ . Furthermore  $\tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij}$  is the numerical flux used to resolve the discontinuity of the predictor solution  $\mathbf{q}_h$  at the space-time sub-face  $\partial C_{ij}^n$ . In the finite volume scheme (42) the flux integral across the quadrilateral sub-surface  $\partial C_{ij}^n$  is computed in an edge-based unit reference system  $(\chi, \tau) \in [0, 1]^2$  that is linked to the physical coordinates of the four space-time nodes that define  $\partial C_{ij}^n$ . Note that in the edge-aligned system the relative time coordinate  $\tau$  is in general *different* from the ones in the adjacent left and right elements  $T_i$  and  $T_j$ , respectively, since the two nodes that define the edge may have already been moved before the update of element  $T_i$ . Let us denote the common edge between element  $T_i$  and  $T_j \in \mathcal{N}_i$  with  $\lambda_{ij}$  and the global number of the first node on  $\lambda_{ij}$  with  $L$  and the one of the second node on the same edge with  $R$ , then the space-time coordinates of the four space-time nodes defining the sub-surface  $\partial C_{ij}^n$  in (42) are given by

$$\tilde{\mathbf{x}}_{ij}^1 = (\mathbf{X}_L^n, t_L^n), \quad \tilde{\mathbf{x}}_{ij}^2 = (\mathbf{X}_R^n, t_R^n), \quad \tilde{\mathbf{x}}_{ij}^3 = (\mathbf{X}_R^{n+1}, t_R^{n+1}), \quad \tilde{\mathbf{x}}_{ij}^4 = (\mathbf{X}_L^{n+1}, t_L^{n+1}). \quad (43)$$

Note that  $L = L(i, j)$  and  $R = R(i, j)$  are functions of the numbers of element  $T_i$  and the neighbor  $T_j$ , respectively, but to ease notation this explicit dependency is dropped. The associated space-time integral of the numerical flux over  $\partial C_{ij}^n$  is also called *edge flux* and denoted by  $\mathbf{G}_{ij}^n$  in the following. The physical times of the four space-time nodes (43) have then to be rescaled to each individual reference space-time coordinate system associated with element  $T_i$  and its neighbor  $T_j$ , respectively, using the time transformation (16).

In order to obtain a conservative scheme, the task of the memory variable  $\mathbf{Q}_i^M$  in (42) is to accumulate (sum) all past fluxes through the lateral space-time sub-surfaces, from the current element time  $t_i^n$  to the current local node times

$t_L^n$  and  $t_R^n$ , respectively, see [36]. The edge flux  $\mathbf{G}_{ij}^n$  through the sub-surface  $\partial C_{ij}^n$  is given by

$$\mathbf{G}_{ij}^n = \int_{\partial C_{ij}^n} \tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} dS = \int_0^1 \int_0^1 |\partial C_{ij}^n| \tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} d\tau d\chi. \quad (44)$$

Then, if element  $T_i$  is updated according to (42), the memory variable of the element itself is reset to zero and the memory variables of the *neighbor* elements  $T_j \in \mathcal{N}_i$  are updated by summing (accumulating) the contribution of the edge-flux  $\mathbf{G}_{ij}^n$  to  $\mathbf{Q}_j^M$ . Note that for element  $T_i$  the contribution  $\mathbf{G}_{ij}^n$  has negative sign. Like in the 1D case presented in [36] we therefore have after each update of element  $T_i$ :

$$\mathbf{Q}_i^M := 0, \quad \mathbf{Q}_j^M := \mathbf{Q}_j^M + \mathbf{G}_{ij}^n, \quad \forall T_j \in \mathcal{N}_i. \quad (45)$$

The implementation of the finite volume scheme (42) requires that a numerical flux is specified through an approximate Riemann solver. A possible simple formulation for the numerical flux is given by the Rusanov-type ALE flux, which, according to [10], reads

$$\tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = \frac{1}{2} (\tilde{\mathbf{F}}(\mathbf{q}_h^+) + \tilde{\mathbf{F}}(\mathbf{q}_h^-)) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} s_{\max} (\mathbf{q}_h^+ - \mathbf{q}_h^-), \quad (46)$$

where  $s_{\max}$  is the maximum eigenvalue of the ALE Jacobian matrix w.r.t. the normal direction in space, which is

$$\mathbf{A}_{\mathbf{n}}^V(\mathbf{Q}) = \left( \sqrt{\tilde{n}_x^2 + \tilde{n}_y^2} \right) \left[ \frac{\partial \mathbf{F}}{\partial \mathbf{Q}} \cdot \mathbf{n} - (\mathbf{V} \cdot \mathbf{n}) \mathbf{I} \right], \quad \mathbf{n} = \frac{(\tilde{n}_x, \tilde{n}_y)^T}{\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2}}, \quad (47)$$

with  $\mathbf{I}$  representing the identity matrix and  $\mathbf{V} \cdot \mathbf{n}$  denoting the local normal mesh velocity.

A more sophisticated alternative is given by the Osher-type numerical flux, which guarantees a less dissipative numerical scheme if compared with the Rusanov flux. It has been presented in [43] for the Eulerian case and then extended to moving meshes in multiple space dimensions in [44, 10, 11]. The corresponding numerical flux is given by

$$\tilde{\mathbf{F}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = \frac{1}{2} (\tilde{\mathbf{F}}(\mathbf{q}_h^+) + \tilde{\mathbf{F}}(\mathbf{q}_h^-)) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} \left( \int_0^1 |\mathbf{A}_{\mathbf{n}}^V(\Psi(s))| ds \right) (\mathbf{q}_h^+ - \mathbf{q}_h^-), \quad (48)$$

where a simple straight-line segment path is used to connect the left and the right state across the discontinuity, i.e.

$$\Psi(s) = \mathbf{q}_h^- + s(\mathbf{q}_h^+ - \mathbf{q}_h^-), \quad 0 \leq s \leq 1. \quad (49)$$

According to [43] the integral in (48) is evaluated numerically using Gaussian quadrature. The absolute value of the dissipation matrix in (48) is evaluated as usual as

$$|\mathbf{A}| = \mathbf{R}|\mathbf{\Lambda}|\mathbf{R}^{-1}, \quad |\mathbf{\Lambda}| = \text{diag}(|\lambda_1|, |\lambda_2|, \dots, |\lambda_\nu|), \quad (50)$$

where  $\mathbf{R}$  and  $\mathbf{R}^{-1}$  denote the right eigenvector matrix and its inverse, respectively.

When element  $T_i$  performs its local time update, the geometry of cell  $T_i$  is also updated, because all three vertices of  $T_i$  are moved according to (37). Using the memory variable  $\mathbf{Q}_i^M$  we ensure conservation of the edge-fluxes, since the numerical fluxes computed over the space-time sub-surfaces  $\partial C_{ij}$  are immediately saved (with opposite sign) in the memory variables of the neighbor elements  $T_j \in \mathcal{N}_i$ . While the consideration of edge fluxes is sufficient for the Lagrangian LTS algorithm presented in [36], its extension to moving unstructured triangular meshes requires an important modification due to the increased topological complexity of a two-dimensional mesh. As shown in Figure 2, each vertex  $k$  of element  $T_i$  is shared among the Voronoi neighbors  $T_j \in \mathcal{V}_k$ . Hence, we must also compute a numerical flux  $\mathbf{G}_{k,m}$  across each edge defined by the vertices  $k$  and  $m$  which does *not* belong to element  $T_i$ , i.e.

$$\mathbf{G}_{k,m} = \int_{\partial \Omega_{k,m}} \tilde{\mathbf{F}}_{l,r} \cdot \tilde{\mathbf{n}}_{l,r} d\tilde{\mathbf{x}}. \quad (51)$$

This *vertex flux* will also be stored (with the proper sign) in the corresponding memory variables  $\mathbf{Q}_l^M$  and  $\mathbf{Q}_r^M$  of elements  $T_l$  and  $T_r$ , where  $l$  denotes the left element and  $r$  denotes the right element on the corresponding edge composed of vertices  $k - m$ , respectively. As shown in Figure 2, the numerical flux is integrated over the *triangular* space-time surfaces  $\Omega_{j,j+1}$ , defined by vertices  $(\tilde{\mathbf{x}}(k), \tilde{\mathbf{x}}(k'), \tilde{\mathbf{x}}(m))$ , that represent the space-time coordinates of vertex  $k$  at the old and at the new time level, and the space-time location of vertex  $k_{i,j}$ , respectively.

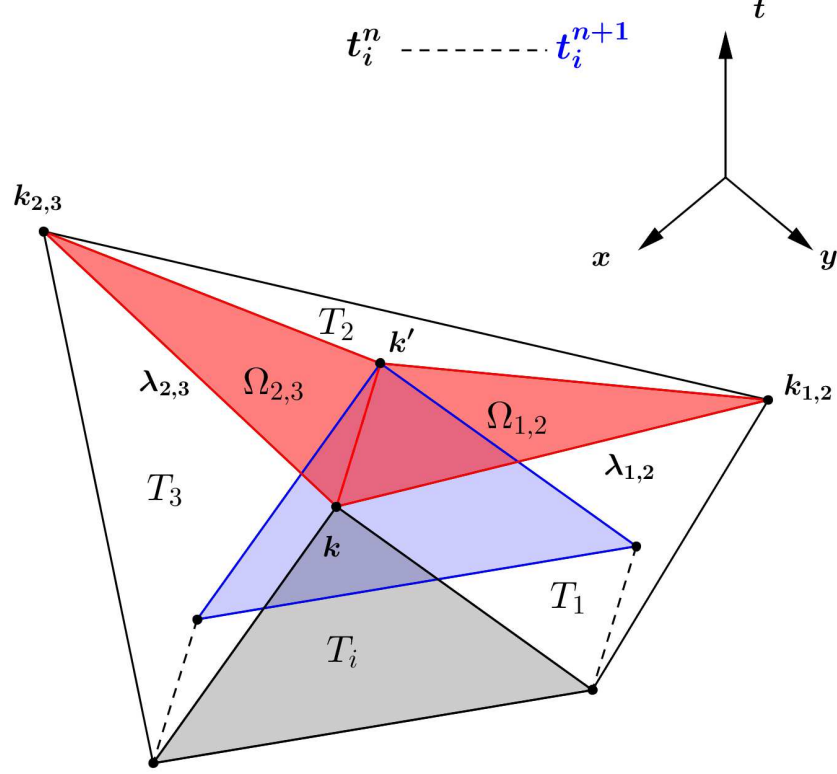


Figure 2: Space-time evolution of element  $T_i$  from time  $t_i^n$  (black triangle) to time  $t_i^{n+1}$  (blue triangle). The triangular sub-surfaces  $\Omega_{1,2}$  and  $\Omega_{2,3}$  are highlighted in red.

In our finite volume formulation we are carrying out an integration over the closed space-time control volume  $C_i^n$ , which automatically guarantees the compliance with the geometric conservation law (GCL), see the appendix of [11] for more details. From the Gauss theorem one has indeed

$$\int_{\partial C_i^n} \tilde{\mathbf{n}} dS = 0. \quad (52)$$

In order to verify whether the GCL is also satisfied in the practical implementation of our Lagrangian LTS algorithm, we need to compute the integral above whenever element  $T_i$  performs an update. For this purpose, we also compute a variable  $H_i^M$  that behaves like the memory variable  $\mathbf{Q}_i^M$ , but for the GCL. All past contributions to the integral (52) relative to the cell  $T_i$  are recorded in the *geometrical memory variable*  $H_i^M$ , which is reset to zero when the local timestep procedure has been completed by element  $T_i$ . Strictly speaking this is not needed, since Eqn. (52) is always satisfied at the end of a local time step because the final space-time control volume is always closed! In all test problems reported in Section 3, property (52) has always been explicitly verified for each element and for each local time step up to machine precision.

### 2.6. Description of the high order Lagrangian LTS algorithm in multiple space dimensions

The aim of this Section is to give an overall overview of the entire LTS algorithm that has been previously described in all its parts. By placing each portion of the algorithm in a context, this presentation should clarify how the numerical scheme can be practically implemented. Due to the LTS approach, where elements are updated in the order given by the update criterion (8), we can no longer speak of *timesteps* but we have to consider *cycles*, as done in [36]. In each cycle the scheme runs over all elements and only those which obey condition (8) are allowed to update the numerical solution, while the others are simply skipped to the next cycle. In the *pre-processing phase* all elements of the mesh

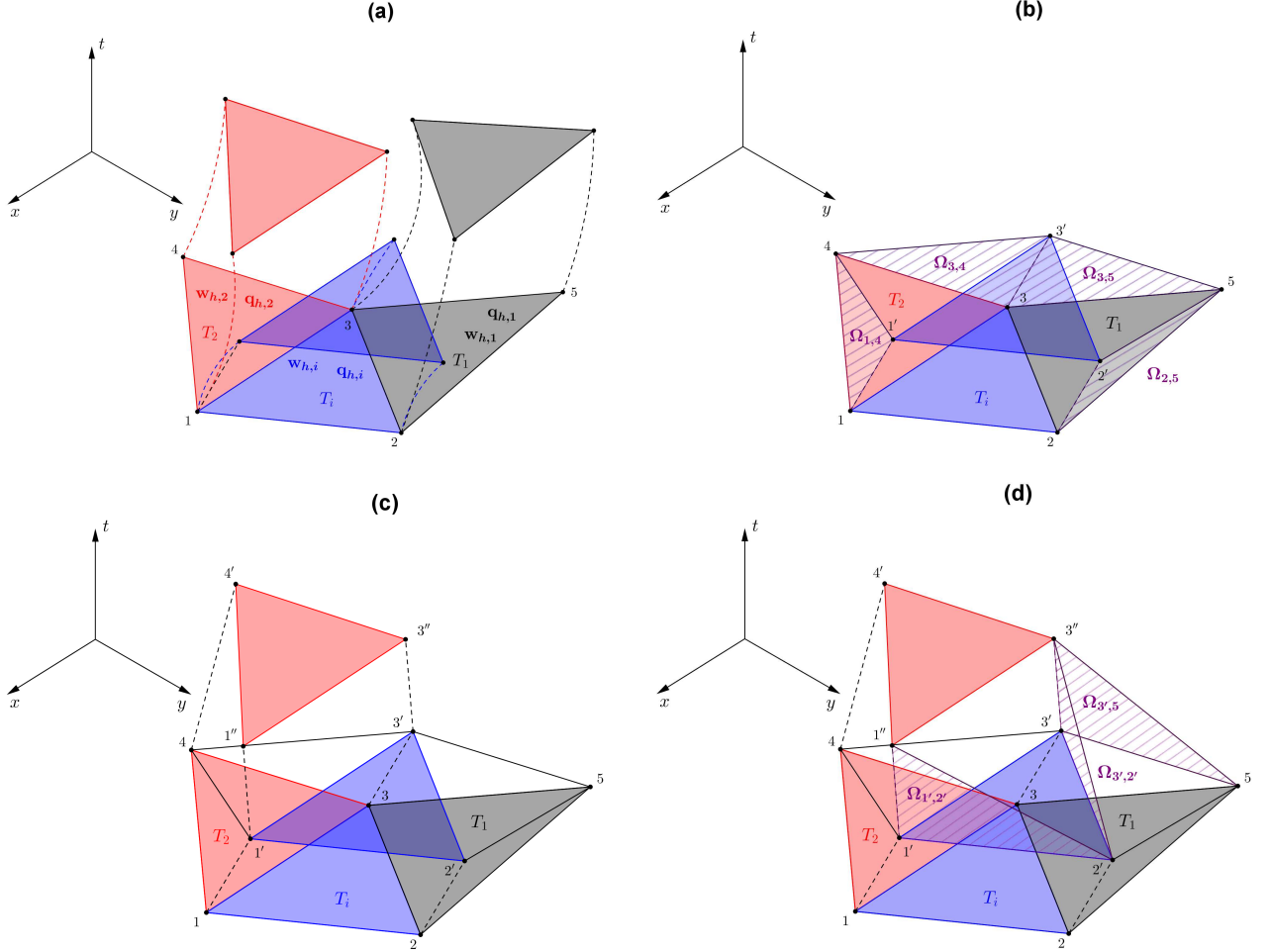


Figure 3: Update of element  $T_i$  and  $T_2$  according to the high order Lagrangian LTS algorithm presented in this paper. At the beginning we assume the *same* current time for each element, i.e.  $t_i^n = t_1^n = t_2^n = t$ . (a) At the current time level  $t$  each element is given its own reconstruction and predictor solution  $\mathbf{w}_h$  and  $\mathbf{q}_h$ , respectively. (b) Update of element  $T_i$  to the new time level  $t_i^{n+1}$ . Computation of the necessary edge fluxes with the direct neighbors and computation of the associated vertex fluxes  $\Omega_{1,4}, \Omega_{3,4}, \Omega_{3,5}, \Omega_{2,5}$ . (c) Update of element  $T_2$ , where the edge fluxes are evaluated only over the space-time surfaces that exceeds the vertex fluxes previously calculated and stored in the memory variable  $\mathbf{Q}_2^M$ . (d) Computation of the vertex fluxes related to the update of  $T_2$ .

are assigned with the initial condition of the problem at the common time level  $t = 0$ , i.e. the cell averages  $\mathbf{Q}_i^n$  are defined according to (4) from the known initial condition. For each element the *first* WENO reconstruction procedure presented in Section 2.2 is carried out. Since all elements are at the same time  $t = 0$ , for this first reconstruction no virtual geometry or virtual cell averages  $\tilde{\mathbf{Q}}$  are needed. As a result, we obtain the high order spatial polynomial  $\mathbf{w}_h$  for each element. Then, the *element-local timestep*  $\Delta t_i^n$  is computed for each cell  $T_i$  according to a classical CFL stability

condition, considering only cell number  $i$  and its Neumann neighborhood  $\mathcal{N}_i$ , i.e.

$$\Delta t_i^n = \min \left( \text{CFL} \frac{\tilde{d}_i}{|\tilde{\lambda}_{\max,i}|}, \text{CFL} \frac{\tilde{d}_j}{|\tilde{\lambda}_{\max,j}|} \right), \quad \forall T_j \in \mathcal{N}_i, \quad (53)$$

with  $\tilde{d}_j = d_j^0$  denoting the incircle diameter of element  $T_j$  and  $|\tilde{\lambda}_{\max,j}| = |\lambda_{\max,j}|^0$  representing the maximum absolute value of the eigenvalues computed from the initial condition  $\tilde{\mathbf{Q}}_j = \mathbf{Q}_j^0$  in  $T_j$ . CFL is the Courant-Friedrichs-Levy number that must satisfy the inequality  $\text{CFL} \leq 0.5$  in the two-dimensional case, as stated in [87]. In the last part of the pre-processing stage, since the local element timestep  $\Delta t_i^n$  as well as the local reconstruction polynomial  $\mathbf{w}_h$  have already been computed, we are able to carry out the local space-time Galerkin predictor procedure described in Section 2.3, which gives the high order local space-time predictor solution  $\mathbf{q}_h$ . All cells are now at the same current time level  $t = 0$  and for each element  $T_i$  the local predictor solution  $\mathbf{q}_h$ , the local reconstruction polynomial  $\mathbf{w}_h$  and the cell average  $\mathbf{Q}_i^n$  are given (Figure 3 (a)). We underline that also each node  $k$  of the entire computational mesh is assigned the initial time level  $t_k^0 = 0$ .

The algorithm proceeds with the *computational phase*, during which each element  $T_i$  will reach the imposed final time of the simulation  $t = t_f$  in a certain number of necessary cycles, according to its own optimal timestep. The first cycle starts by looping over all elements to check in which elements the update criterion (8) is satisfied. If an element  $T_i$  obeys condition (8), then it performs the local timestep until its future time  $t_i^{n+1} = t_i^n + \Delta t_i^n$  (Figure 3 (b)) through the following sub-steps:

- *mesh motion*: each vertex  $k$  of element  $T_i$  is moved to the new position at time  $t_k^{n+1} = t_i^{n+1}$  using the node solver algorithm illustrated in Section 2.4 and all other geometric quantities of element  $T_i$  are also updated;
- *edge flux computation*: we compute the numerical fluxes  $\mathbf{G}_{ij}^n$  through the quadrilateral space-time sub-surfaces and using the high order Lagrangian finite volume scheme (42) we obtain the numerical solution  $\mathbf{Q}_i^{n+1}$ . Subsequently, we reset the memory variable of element  $T_i$  to zero, i.e.  $\mathbf{Q}_i^M := 0$  and accumulate the edge-fluxes into the memory variables of the neighbor elements to maintain conservation ( $\mathbf{Q}_j^M := \mathbf{Q}_j^M + \mathbf{G}_{ij}^n$ ). Also the geometry variable  $H_i^M$  is reset to zero, after assuring that condition (52) is satisfied;
- *vertex flux computation*: as explained in Section 2.5, for each vertex  $k$  of the element  $T_i$  we also need to evaluate for each edge  $k - k_{i,j}$  the additional fluxes  $\mathbf{G}_{k,j}$  using (51) (Figure 3 (b)). The numerical fluxes evaluated over the space-time triangular sub-surface  $\Omega_{j,j+1}$  (see Figure 2) are immediately stored into the memory variable of the adjacent elements  $T_j, T_{j+1}$ , while the part of the geometry integral (52) is stored into  $H_j^M$  and  $H_{j+1}^M$ . In this way we ensure that the numerical scheme is fully conservative;
- *virtual projection*: all the elements  $T_j$  belonging to the entire reconstruction stencil  $\mathbf{S}_i^W$  of element  $T_i$  are now moved *virtually* to the future time level of cell  $i$ , i.e.  $t_i^{n+1}$ , and also the virtual cell averages  $\tilde{\mathbf{Q}}_j$  are *estimated* from the local predictor solution  $\mathbf{q}_h$  in the neighbors  $T_j$ ;
- *local WENO reconstruction*: once the *virtual* geometry and cell averages have been projected to the future time  $t_i^{n+1}$ , the local WENO reconstruction technique described in Section 2.2 can be carried out for element  $T_i$ , hence obtaining the new reconstruction polynomial  $\mathbf{w}_h$  at time  $t_i^{n+1}$ ;
- *local timestep computation*: using the virtual geometry and the virtual solution of the Neumann neighbors, the next local timestep  $\Delta t_i^{n+1}$  is evaluated according to (53);
- *local space-time predictor*: finally we compute the high order space-time predictor solution  $\mathbf{q}_h$  valid within the next timestep of element  $T_i$ .

This procedure is repeated for all elements, until all of them reach the final time of the simulation  $t_f$ . As soon as an element  $T_i$  has finished its own computation because it has reached the final time  $t_f$ , it is automatically skipped at the beginning of each cycle, waiting for the remaining elements to reach the final time, too.

This brief description summarizes how our high order Lagrangian LTS algorithm is organized. During the simulation hanging nodes in time appear because each node is moved *physically* only by the updating element  $T_i$  which



the vertex belongs to. As a consequence, the resulting space-time mesh is computed *dynamically*, producing a *non-conforming* space-time mesh. Due to our high order approach, the edge and vertex fluxes have to be evaluated using higher order Gaussian quadrature rules, hence increasing the computational cost. In practical applications, for which first or second order accurate finite volume schemes are considered adequate, one could rely on the fast and simple mid-point rule that would significantly improve the computational efficiency of our LTS algorithm.

### 3. Test problems

In the following we solve some numerical test problems in order to validate the high order Lagrangian ADER-WENO algorithm with time accurate local time stepping (LTS) presented so far. We consider the two-dimensional Euler equations of compressible gas dynamics, which can be cast into form (1) with

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \end{pmatrix}, \quad (54)$$

where the vector of conserved variables is denoted by  $\mathbf{Q}$  and the flux tensor is addressed with  $\mathbf{F} = (\mathbf{f}, \mathbf{g})$ . Furthermore let  $\rho$  and  $\rho E$  denote the mass density and the total energy density, respectively, while  $\mathbf{v} = (u, v)$  represents the velocity vector and  $p$  is the fluid pressure. The source term  $\mathbf{S}(\mathbf{Q})$  is zero for the homogeneous Euler equations. The system is closed using the equation of state (EOS) for an ideal gas, namely

$$p = (\gamma - 1) \left( \rho E - \frac{1}{2} \rho (u^2 + v^2) \right), \quad (55)$$

where  $\gamma$  is the ratio of specific heats.

In the next sections the governing PDE (1), with the definitions provided by (54), will be assigned with different initial conditions, that may be given either in terms of the vector of conserved variables  $\mathbf{Q} = (\rho, \rho u, \rho v, \rho E)$  or of the primitive variables  $\mathcal{U} = (\rho, u, v, p)$ . The system will be solved applying the Lagrangian ADER-WENO finite volume schemes illustrated in Section 2.5, choosing among the Rusanov-type (46) and the Osher-type (48) numerical fluxes. In all the proposed test problems the local mesh velocity is chosen to be equal to the local fluid velocity ( $\mathbf{V} = \mathbf{v}$ ), hence a formulation of our ALE algorithm has been chosen that comes as close as possible to a truly Lagrangian scheme.

#### 3.1. Numerical convergence studies

In order to carry out the numerical convergence studies for the high order LTS Lagrangian schemes we consider the classical smooth convected isentropic vortex proposed on triangular grids by Hu and Shu [57]. The initial computational domain is the square  $\Omega(0) = [0; 10] \times [0; 10]$  defined on the  $\mathbf{x} = (x, y)$  plane with periodic boundary conditions imposed on each side. The initial condition is given in terms of primitive variables as a linear superposition of a homogeneous background field and a perturbation:

$$\mathcal{U} = (\rho, u, v, p) = (1 + \delta\rho, 1 + \delta u, 1 + \delta v, 1 + \delta p). \quad (56)$$

The flow is assumed to be isentropic, hence with no perturbation in the entropy, while the perturbations for velocity  $\mathbf{v} = (u, v)$  and temperature  $T$  are given by

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \frac{\epsilon}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y-5) \\ (x-5) \end{pmatrix}, \quad \delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}, \quad (57)$$

where  $r^2 = (x-5)^2 + (y-5)^2$  is the vortex radius,  $\epsilon = 5$  denotes the vortex strength and the ratio of specific heats is set to  $\gamma = 1.4$ . The perturbations for density and pressure are then expressed as

$$\delta\rho = (1 + \delta T)^{\frac{1}{\gamma-1}} - 1, \quad \delta p = (1 + \delta T)^{\frac{\gamma}{\gamma-1}} - 1. \quad (58)$$

Table 1: Numerical convergence results for the compressible Euler equations using second to fourth order Lagrangian ADER-WENO finite volume schemes with time accurate local time stepping (LTS). The error norms refer to the variable  $\rho$  (density) at time  $t = 1.0$ .

$O2$			$O3$			$O4$		
$h(\Omega(t_f))$	$\epsilon_{L_2}$	$O(L_2)$	$h(\Omega, t_f)$	$\epsilon_{L_2}$	$O(L_2)$	$h(\Omega(t_f))$	$\epsilon_{L_2}$	$O(L_2)$
3.58E-01	5.286E-02	-	3.32E-01	3.471E-02	-	7.00E-01	6.419E-02	-
2.48E-01	3.558E-02	1.1	2.51E-01	1.789E-02	2.4	3.28E-01	1.030E-02	2.4
1.70E-01	1.514E-02	2.3	1.68E-01	6.346E-03	2.6	2.51E-01	3.598E-03	3.9
1.28E-01	8.193E-03	2.1	1.28E-01	2.935E-03	2.8	1.68E-01	7.706E-04	3.8

The vortex is convected with velocity  $\mathbf{v}_c = (1, 1)$ , so that at the final time  $t_f$  of the simulation the exact solution  $\mathbf{Q}_e(\mathbf{x}, t_f)$  is simply given by the time-shifted initial condition, e.g.  $\mathbf{Q}_e(\mathbf{x}, t_f) = \mathbf{Q}(\mathbf{x} - \mathbf{v}_c t_f, 0)$ , with the averaged convection velocity of the vortex  $\mathbf{v}_c = (1, 1)$ . As depicted in Figure 4, the mesh is highly distorted and twisted by the vortex motion and no rezoning algorithm [12, 9] is adopted here because we want to validate the new LTS algorithm inside an almost fully Lagrangian approach. Therefore the final time of the simulation is chosen to be  $t_f = 1.0$ , which allows the computational mesh to remain reasonably well-shaped. We run this test case on successive refined meshes and for each mesh the corresponding error is expressed in the continuous  $L_2$  norm as

$$\epsilon_{L_2} = \sqrt{\int_{\Omega(t_f)} (\mathbf{Q}_e(x, y, t_f) - \mathbf{w}_h(x, y, t_f))^2 dx dy}, \quad (59)$$

where  $\mathbf{w}_h(x, y, t_f)$  represents the high order reconstructed solution at the final time, while the mesh size  $h(\Omega(t_f))$  is evaluated as the maximum diameter of the circumcircles of the triangles in the final computational domain  $\Omega(t_f)$ . We use the Rusanov-type numerical flux (46) to obtain the convergence results listed in Table 1, achieving the designed order of accuracy of the scheme very well.

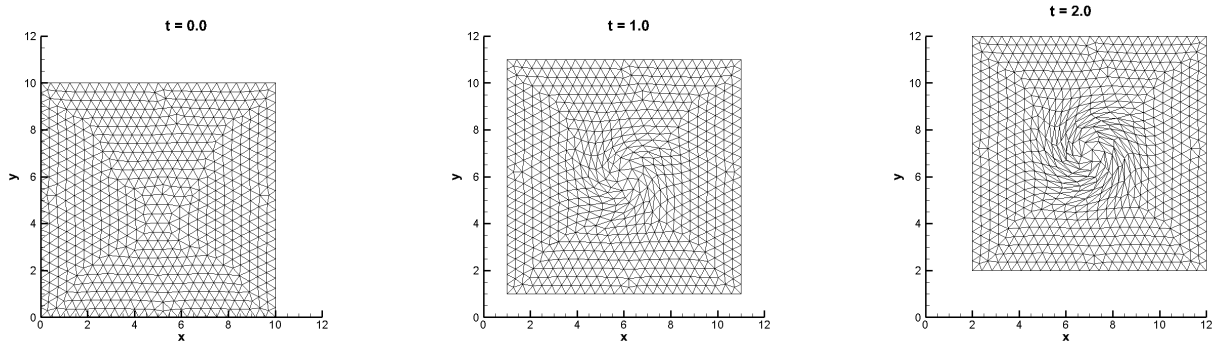


Figure 4: Mesh configuration at three different output times for the smooth isentropic vortex test problem. The mesh is highly twisted in the center of the computational domain, which is furthermore convected with velocity  $\mathbf{v}_c = (1, 1)$ .

### 3.2. Riemann problems

Here we solve two classical Riemann problems, namely the shock tube problems of Sod and the Lax, which are in the following addressed as RP1 and RP2, respectively, and which are widely adopted to validate numerical algorithms for the solution of the compressible Euler equations. They both include the formation of a left-propagating rarefaction wave, an intermediate contact discontinuity and a right-propagating shock wave. Though intrinsically one-dimensional, these tests become non-trivial and multidimensional when applied to unstructured meshes, where in

Table 2: Initial condition for the Sod (RP1) and the Lax (RP2) shock tube problem.  $t_f$  is the final time of the simulation and  $x_d$  denotes the position of the initial discontinuity.

Case	$\rho_L$	$u_L$	$v_L$	$p_L$	$\rho_R$	$u_R$	$v_R$	$p_R$	$t_f$	$x_d$
RP1	1.0	0.0	0.0	1.0	0.125	0.0	0.0	0.1	0.2	0.0
RP2	0.445	0.698	0.0	3.528	0.5	0.0	0.0	0.571	0.1	0.0

general the element edges are not aligned with the fluid motion. Since a contact wave is present in the solution, we can also check how well it is resolved by our Lagrangian LTS scheme.

The initial computational domain is given by the box  $\Omega(0) = [-0.5; 0.5] \times [-0.05; 0.05]$  that is discretized with a characteristic mesh size of  $h = 1/200$ , leading to a total number of  $N_E = 8862$  elements, while the initial conditions are given in terms of the primitive variables  $\mathcal{U} = (\rho, u, v, p)$ . Table 2 reports the relevant data for the setup of the two tests, where  $t_f$  represents the final time of the simulation while  $x_d$  gives the position of the initial discontinuity which splits the computational domain, as well as the initial conditions, in the two left and right states  $\mathcal{U}_L$  and  $\mathcal{U}_R$ . We set periodic boundary conditions in the  $y$  direction, while transmissive boundaries are imposed along the  $x$  direction. The ratio of specific heats is assumed to be  $\gamma = 1.4$  for both Riemann problems.

The exact solution is computed with the exact Riemann solver presented in [87]. We use the third order version of our Lagrangian ADER-WENO schemes with LTS using the Osher-type numerical flux to obtain the results depicted in Figures 5-6, where a comparison between the exact and the numerical solution is shown. We observe an excellent resolution of the contact wave with only one intermediate point for both RP1 and RP2, and a very good agreement with the analytical solution can also be noticed for density, as well as for pressure and for the horizontal velocity component. Table 3 aims at showing the computational efficiency of the LTS algorithm w.r.t. the Lagrangian ADER-WENO schemes with global time stepping (GTS) presented in [10]. In order to give a fair comparison between LTS and GTS schemes, the efficiency is not measured in terms of computational time, which may depend on the machine hardware or on the algorithm implementation, but rather we count the total number of element updates needed to reach the final time of the simulation, as done in [36]. Hence, looking at Table 3, we notice that the Lagrangian algorithm with global time stepping requires a total number of element updates that is a factor of 3-4 times larger than the one of our new Lagrangian scheme with LTS.

Table 3: Comparison of the computational efficiency between GTS and LTS algorithm in terms of the total number of element updates for RP1 and RP2. A third order scheme has been adopted.

Case	Number of element updates		
	GTS	LTS	GTS/LTS
RP1	$10.120404 \cdot 10^6$	$3.257847 \cdot 10^6$	3.11
RP2	$23.349964 \cdot 10^6$	$5.020780 \cdot 10^6$	4.45

### 3.3. Two-dimensional explosion problems

Circular explosion problems can be regarded as the two-dimensional extension of Riemann problems. The initial domain  $\Omega(0) = \{\mathbf{x} : \|\mathbf{x}\| < R_o\}$  is given by the unit circle of radius  $R_o = 1$ . A circle of radius  $R = 0.5$  separates two different states that define the initial conditions reported in Table 4 in terms of primitive variables  $\mathcal{U} = (\rho, u, v, p)$ . The two states are addressed here as the *inner* state  $\mathcal{U}_i$  and the *outer* state  $\mathcal{U}_o$ , respectively. Transmissive boundary conditions have been imposed on the external boundary and we set  $\gamma = 1.4$ . EP1 corresponds to the initial data of the classical Sod shock tube problem RP1, while EP2 is taken from [87]. In both cases we use the same computational mesh  $m_1$ , with a characteristic mesh size of  $h = 1/100$  for  $r \leq R$  and  $h = 1/50$  for  $r > R$ , hence obtaining a total number  $N_E = 43756$  of triangles.

As proposed in [10] a suitable reference solution can be obtained simplifying the two-dimensional Euler equations

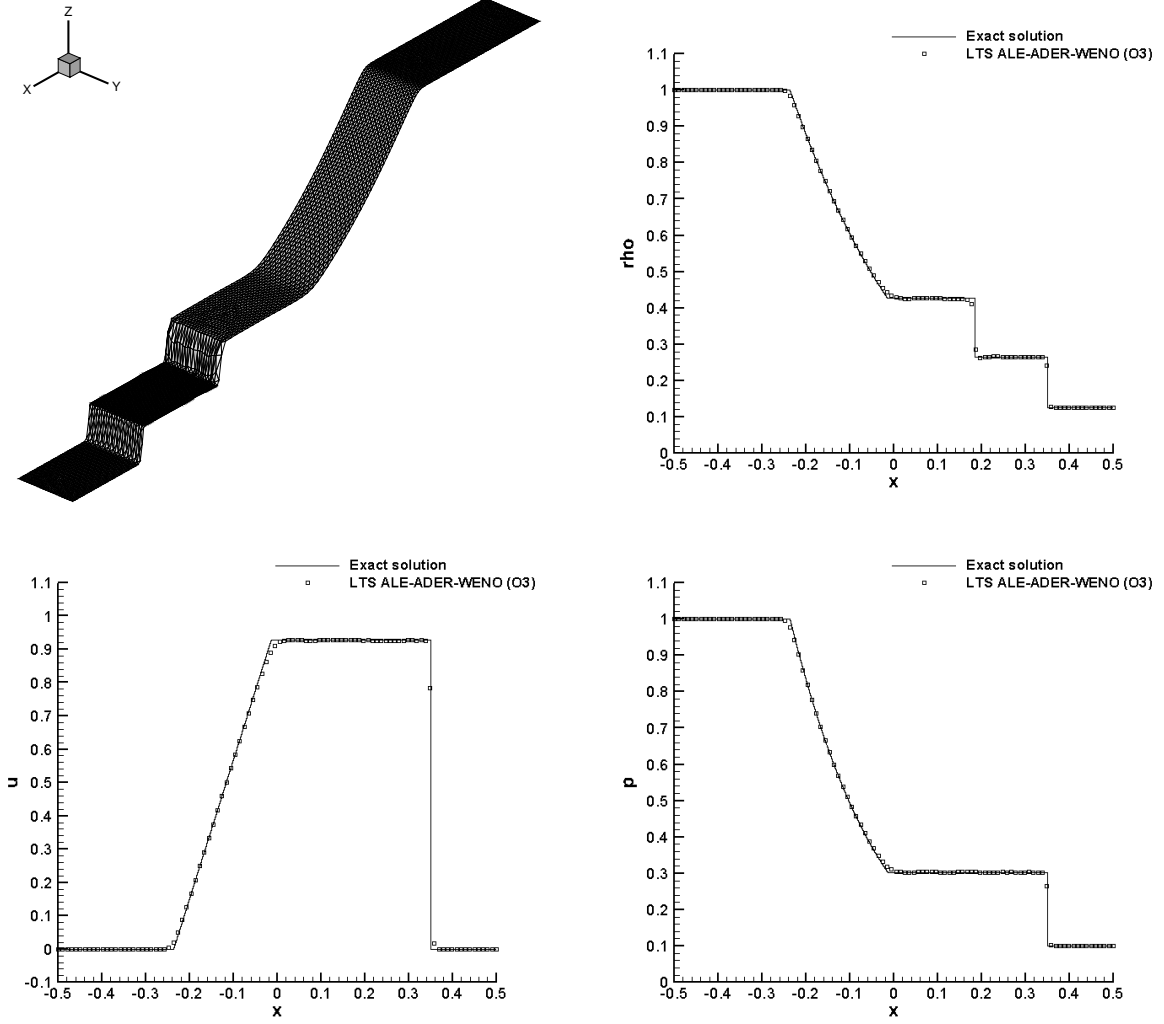


Figure 5: Comparison between exact and third order accurate numerical solution for the Sod shock tube problem RP1. Density (top right), velocity (bottom left) and pressure (bottom right) distribution are shown as well as a 3D view of the density solution at the final time  $t_f = 0.2$  (top left).

Table 4: Initial conditions for the two-dimensional explosion problems EP1 and the EP2 with  $t_f$  denoting the final time of the simulation.

Case	$\rho_i$	$u_i$	$v_i$	$p_i$	$\rho_o$	$u_o$	$v_o$	$p_o$	$t_f$
EP1	1.0	0.0	0.0	1.0	0.125	0.0	0.0	0.1	0.2
EP2	1.0	0.0	0.0	1000.0	1.0	0.0	0.0	0.01	0.012

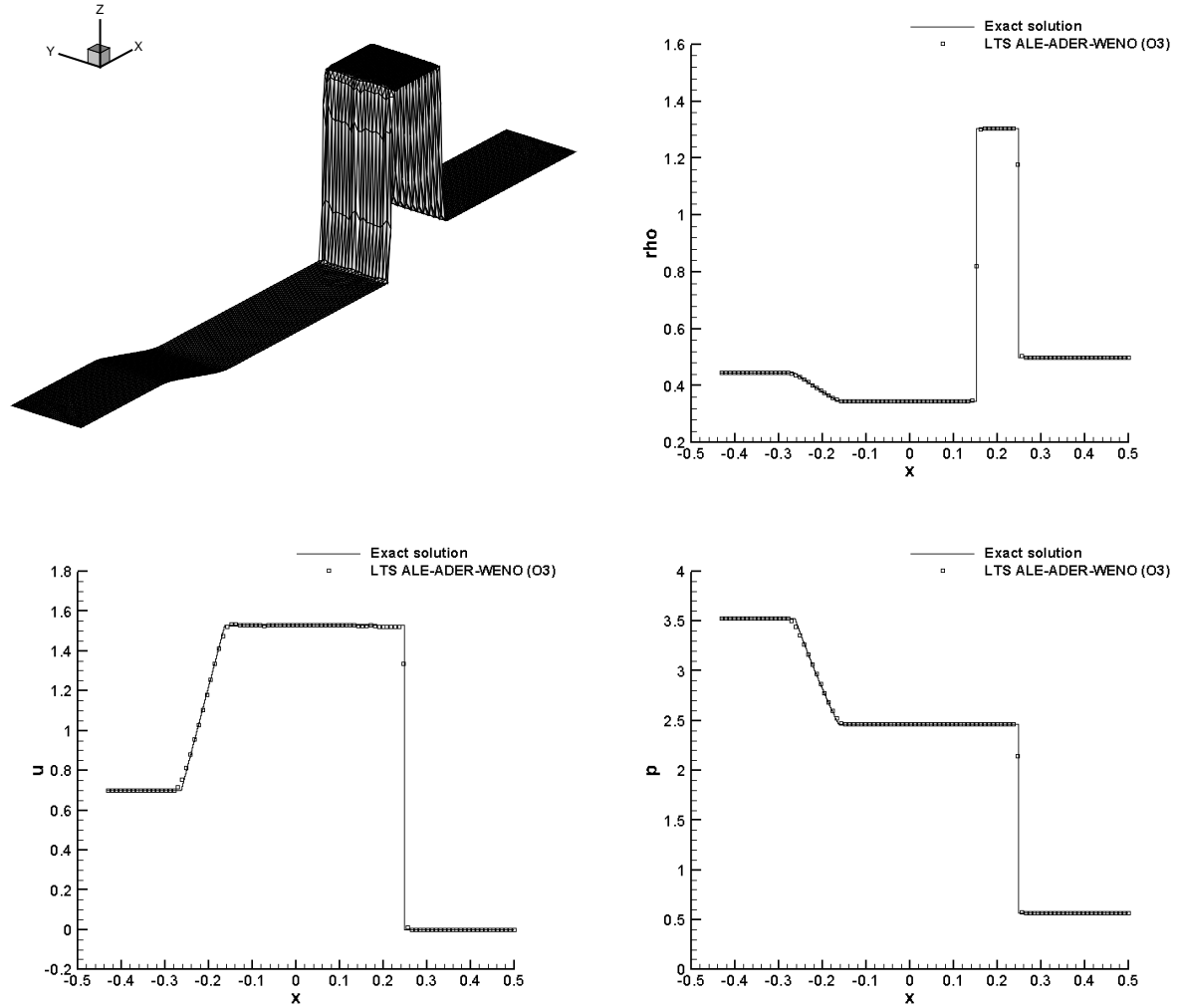


Figure 6: Comparison between exact and third order accurate numerical solution for the Lax shock tube problem RP2. Density (top right), velocity (bottom left) and pressure (bottom right) distribution are shown as well as a 3D view of the density solution at the final time  $t_f = 0.1$  (top left).

to a one-dimensional system with geometric source terms [87], which reads

$$\mathbf{Q}_t + \mathbf{F}(\mathbf{Q})_r = \mathbf{S}(\mathbf{Q}), \quad (60)$$

with

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(\rho E + p) \end{pmatrix}, \quad \mathbf{S} = -\frac{1}{r} \begin{pmatrix} \rho u \\ \rho u^2 \\ u(\rho E + p) \end{pmatrix}. \quad (61)$$

Here  $r$  and  $u$  represent the radial direction and the radial velocity, respectively. As a result, a proper reference solution is obtained after solving the inhomogeneous system of equations (60)-(61) on a one-dimensional mesh of 15000 points in the radial interval  $r \in [0; 1]$  using a classical second order TVD scheme [87] with a Rusanov-type numerical flux. Third order accurate Lagrangian ADER-WENO schemes with time accurate local time stepping have been used together with the Osher-type numerical flux (48) to compute the explosion problems EP1 and EP2. Figures 7-8 show a comparison between the numerical solution obtained with the Lagrangian LTS scheme and the 1D reference solution. As for the Riemann problems presented in the previous section, one can appreciate the very good resolution of the contact wave in the density distribution and a good agreement with the reference solution is achieved also for horizontal velocity and pressure. We point out that EP2 is more challenging than EP1 because it involves a strong shock wave which causes a high compression of some elements in the mesh, as clearly depicted in Figure 9. By using the LTS approach we can avoid that those small triangles dictate the timestep for the entire mesh, hence allowing the other control volumes to reach the end of the simulation much faster and with a lower number of element updates, as highlighted in Table 5.

The numerical simulation of many important phenomena arising in science and engineering typically requires the use of non-uniform computational grids with small elements clustered in some portions of the computational domain. In such circumstances, the use of a classical global time stepping algorithm would slow down the computation severely, since the smallest element of the mesh reduces the admissible timestep for the entire grid. Within the Eulerian framework on Cartesian grids, such a problem can be conveniently circumvented by resorting to Adaptive-Mesh-Refinement (AMR) with local time stepping, see e.g. [8, 7, 4, 3, 14, 45, 39, 91]. There, the mesh is forced to refine only when and where this is needed, while it is recoarsened as soon as the chosen refinement criterion is no longer satisfied. An alternative option consists of preparing the computational mesh with a *local static* refinement, which will remain fixed during the evolution if an Eulerian approach is adopted, while it will respond to the dynamics of the fluid if a Lagrangian framework is adopted, like in the present paper. In both cases, a local time stepping algorithm would make a huge difference in terms of computational efficiency, avoiding large control volumes to be slowed down by very small ones. Motivated by these considerations, we have run a modified version of the explosion problem EP1, denoted as EP1\*, which uses the same initial conditions of the former, apart for the mesh  $m_2$ , which has been built with a local mesh refinement around the initial location of the discontinuity, i.e. at  $R = 0.5$ . More specifically, the mesh size is  $h = 1/100$  in the refined zone, and it grows with a growth rate of  $s = 1.5$  until  $h = 1/10$ , which is used in the rest of the domain. Figure 10 shows the initial and the final configuration of the computational grid  $m_2$  as well as a zoom onto the discontinuity. With this test problem we want to make another case for adopting the LTS approach rather than the classical GTS algorithm, and the advantages of the former can be easily deduced by looking at Table 5, where we compare the total number of element updates for each explosion problem needed to reach the final time of the simulation.

Table 5: Comparison of the computational efficiency between GTS and LTS algorithm using the total number of element updates for EP1, EP2 and EP1\*.

Case	Number of element updates		
	GTS	LTS	GTS/LTS
EP1	$30.804224 \cdot 10^6$	$12.206887 \cdot 10^6$	2.52
EP2	$181.412376 \cdot 10^6$	$38.274477 \cdot 10^6$	4.74
EP1*	$22.171520 \cdot 10^6$	$8.313123 \cdot 10^6$	2.67



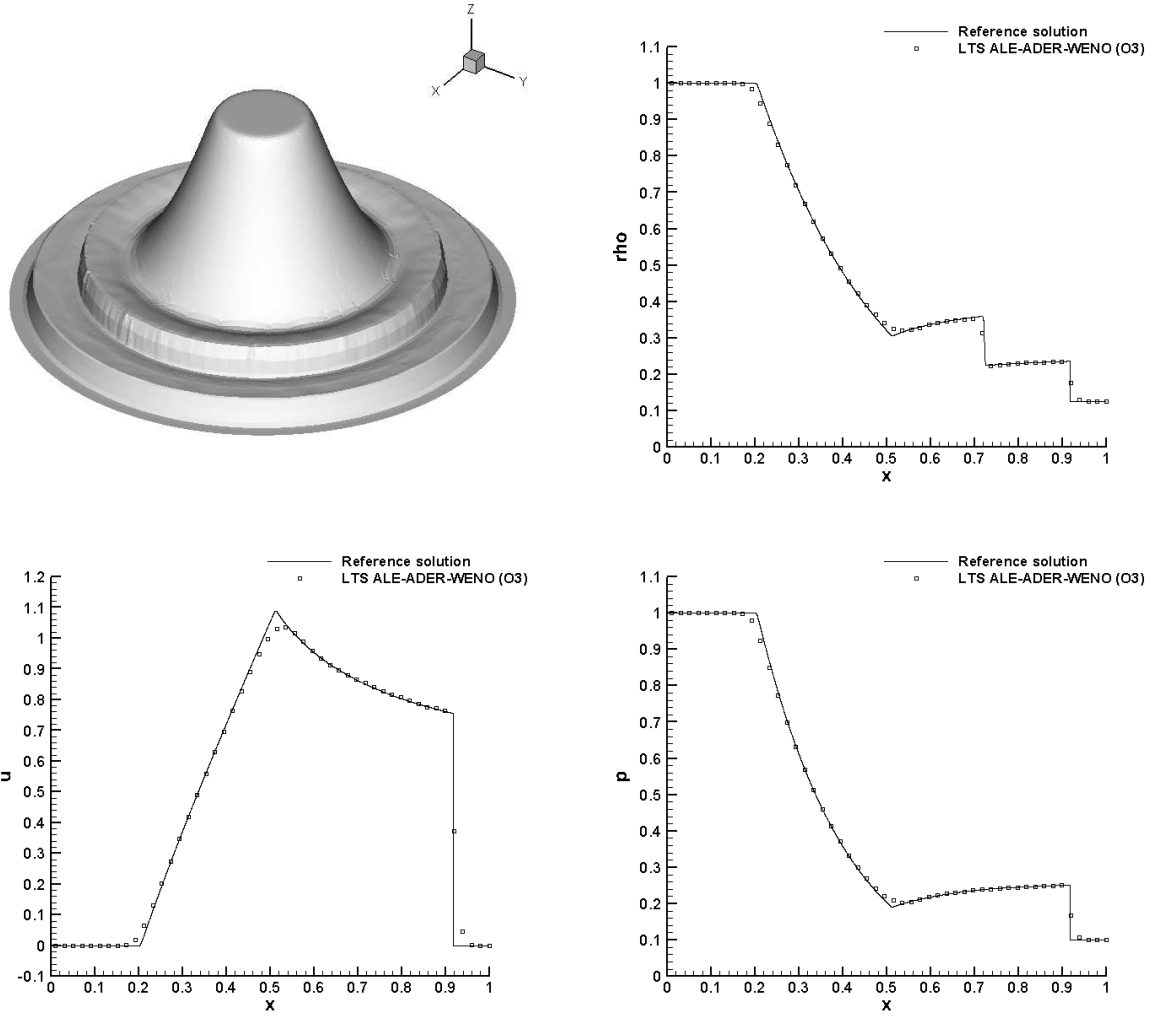


Figure 7: Comparison between reference and third order accurate numerical solution for the explosion problem EP1. Density (top right), velocity (bottom left) and pressure (bottom right) distribution are shown as well as a 3D view of the density solution at the final time  $t_f = 0.25$  (top left).

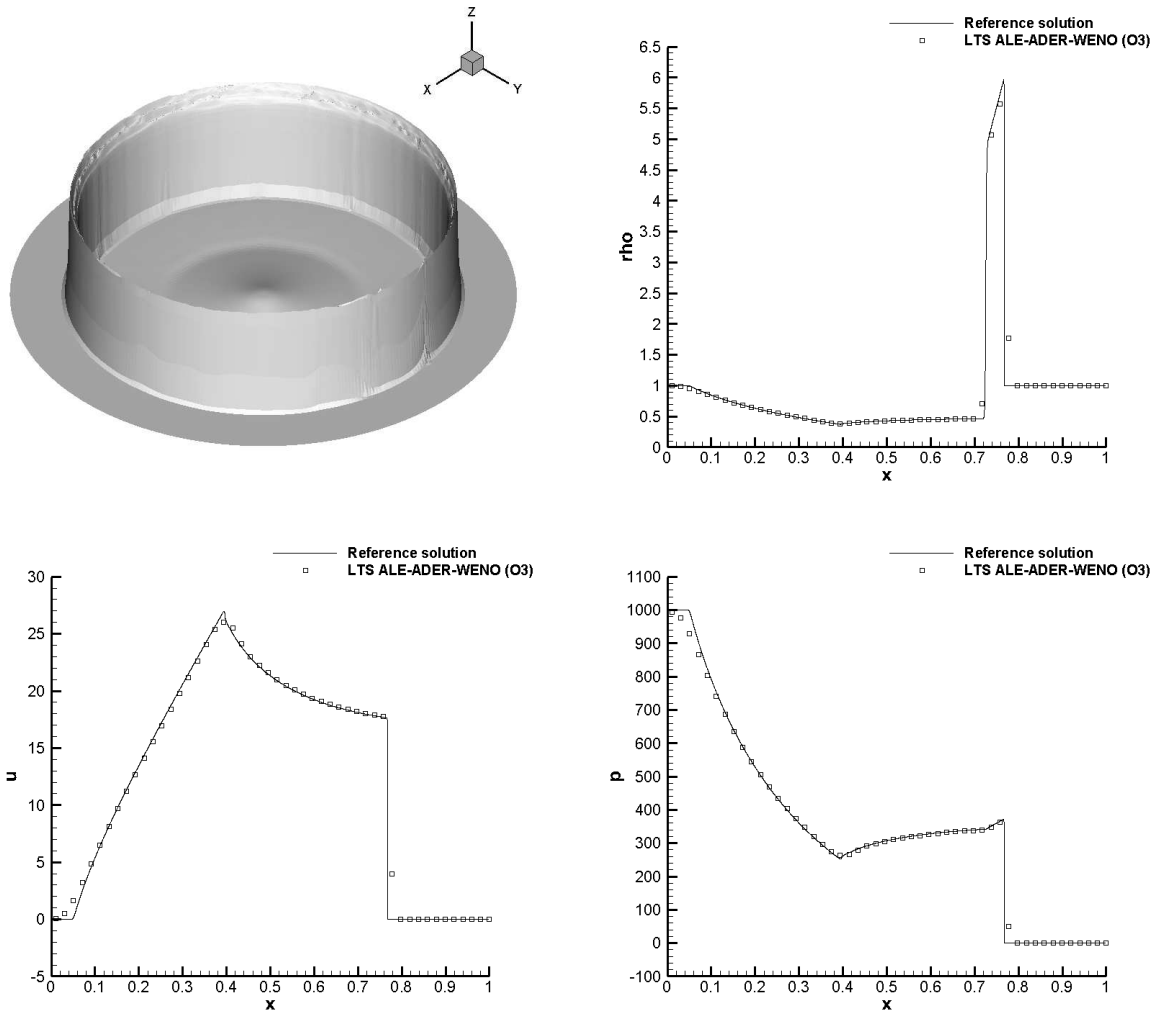


Figure 8: Comparison between reference and third order accurate numerical solution for the explosion problem EP2. Density (top right), velocity (bottom left) and pressure (bottom right) distribution are shown as well as a 3D view of the density solution at the final time  $t_f = 0.012$  (top left).

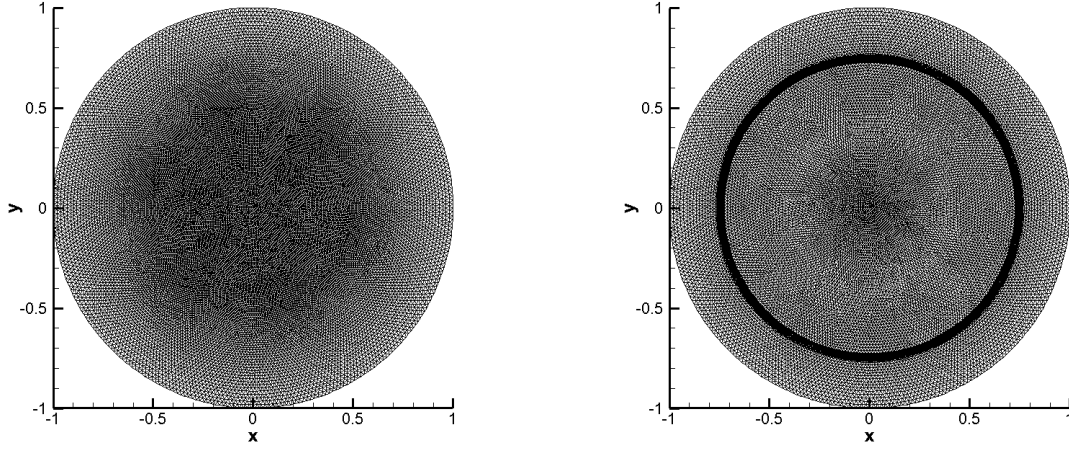


Figure 9: Initial (left) and final (right) mesh configuration for the explosion problem EP2. The strong shock generates a high compression of those elements which follow the wave.

### 3.4. The Kidder problem

The Kidder problem is a classical benchmark problem for Lagrangian algorithms. It has been widely used in the literature [69, 15] in order to assure that no spurious entropy is produced by the Lagrangian scheme. This test case was first designed by Kidder in [61] and it consists of an isentropic compression of a portion of a shell filled with an ideal gas. The shell  $\Omega(0)$  is initially bounded by  $r_i(t) \leq r \leq r_e(t)$ , where  $r = \sqrt{x^2 + y^2}$  represents the general radial coordinate while  $r_i(t), r_e(t)$  denote the time-dependent internal and external radius, respectively. The perfect gas is initially assigned with the following vector of primitive variables  $\mathcal{U}_0$ :

$$\mathcal{U}_0 = \begin{pmatrix} \rho_0(r) \\ u_0(r) \\ v_0(r) \\ p_0(r) \end{pmatrix} = \begin{pmatrix} \left( \frac{r_{e,0}^2 - r^2}{r_{e,0}^2 - r_{i,0}^2} \rho_{i,0}^{\gamma-1} + \frac{r^2 - r_{i,0}^2}{r_{e,0}^2 - r_{e,0}^2} \rho_{e,0}^{\gamma-1} \right)^{\frac{1}{\gamma-1}} \\ 0 \\ 0 \\ s_0 \rho_0(r)^\gamma \end{pmatrix}, \quad (62)$$

where  $\rho_{i,0} = 1$  and  $\rho_{e,0} = 2$  are the initial values of density at the internal and the external frontier, respectively. According to [69] the ratio of specific heats is  $\gamma = 2$  and the initial entropy distribution  $s_0$  is assumed to be uniform, i.e.  $s_0 = \frac{p_0}{\rho_0} = 1$ .

Sliding wall boundary conditions are imposed on the horizontal and vertical edges that bound the portion of the shell, while on the internal and on the external frontier we set a space-time dependent state, which is assigned according to the exact solution  $R(r, t)$  [61]. The analytical solution for the Kidder problem is given at the general time  $t$  for a fluid particle initially located at radius  $r$  as a function of the radius and of the homothety rate  $h(t)$ , i.e.

$$R(r, t) = h(t)r, \quad h(t) = \sqrt{1 - \frac{t^2}{\tau^2}}, \quad (63)$$

where  $\tau$  is the focalisation time

$$\tau = \sqrt{\frac{\gamma - 1}{2} \frac{(r_{e,0}^2 - r_{i,0}^2)}{c_{e,0}^2 - c_{i,0}^2}} \quad (64)$$

with  $c_{i,e} = \sqrt{\gamma \frac{p_{i,e}}{\rho_{i,e}}}$  representing the internal and external sound speeds. Following [15, 69], the final time of the simulation is chosen to be  $t_f = \frac{\sqrt{3}}{2}\tau$ , so that the compression rate is  $h(t_f) = 0.5$  and the exact solution is given by

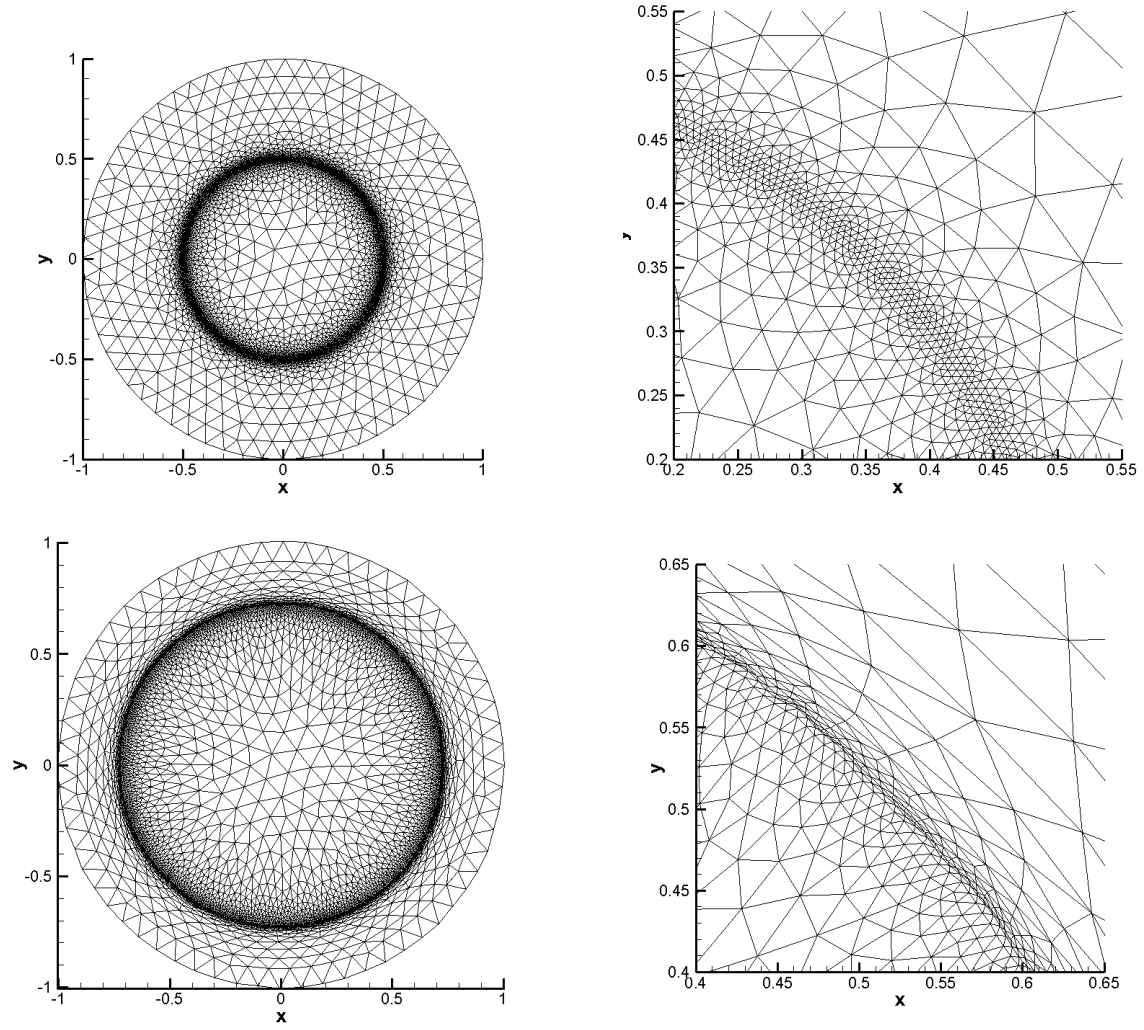


Figure 10: Initial (top) and final (bottom) configuration of the grid  $m_2$  for the explosion problem EP1\*. A zoom of the mesh configuration across the contact wave is shown on the right at time  $t = 0.0$  (top) and  $t = 0.25$  (bottom).

the shell located within the interval  $0.45 \leq R \leq 0.5$ . We use a fourth order accurate version of our new Lagrangian ADER-WENO scheme with LTS using the Osher-type numerical flux (48). The results are depicted in Figure 11, which shows the numerical solution for density at three different output times  $t = 0.0$ ,  $t = 0.9$  and  $t = t_f$ . Moreover, the evolution of the internal and external radius of the shell has been monitored during the simulation and Table 6 reports the absolute error  $|err|$  of the frontier positions, which is defined as the difference between the analytical and the numerical location of the internal and external radius at the final time.

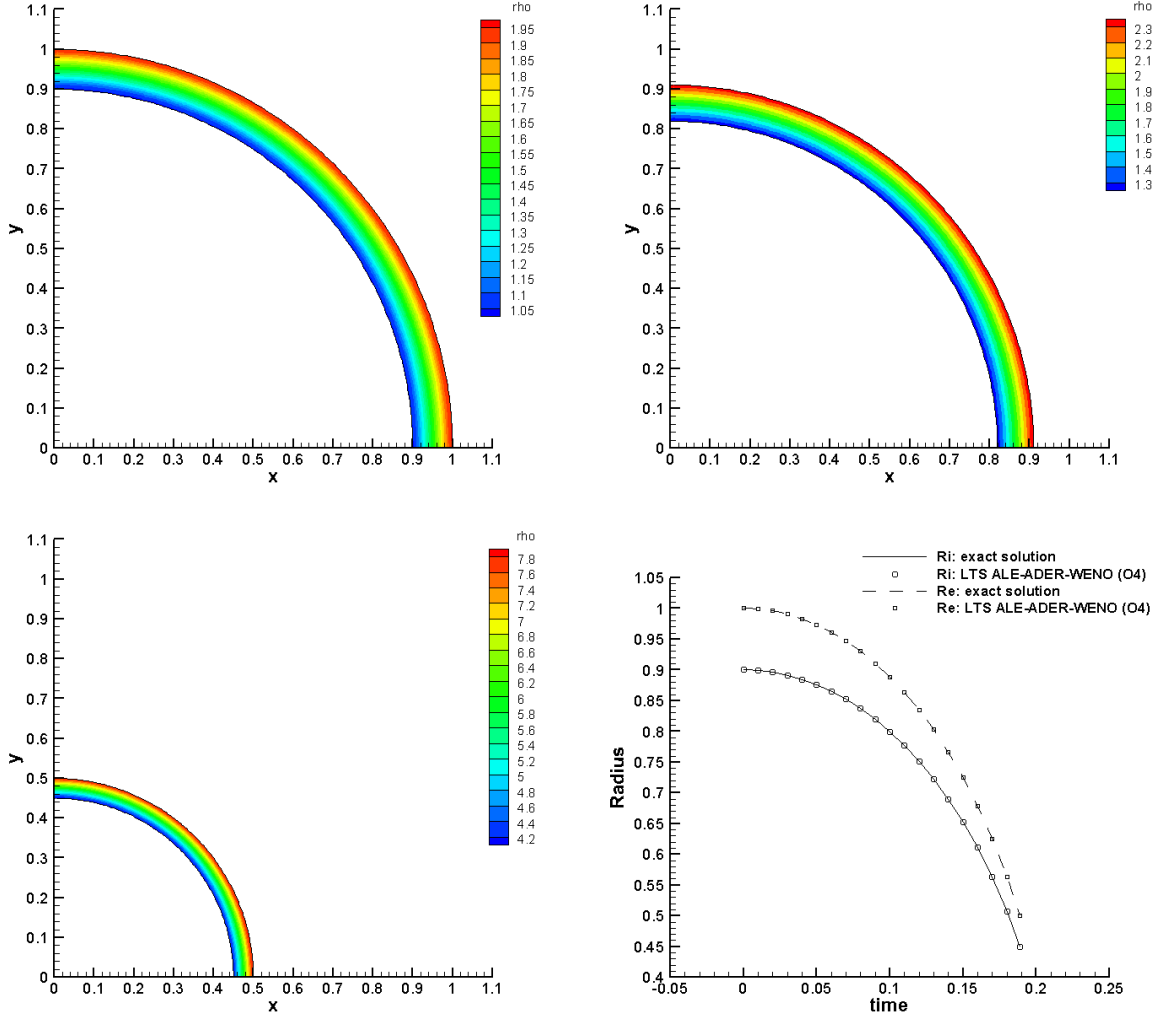


Figure 11: Fourth order accurate density distribution for the Kidder problem at the initial time  $t = 0.0$  (top left), at  $t = 0.9$  (top right) and at the final time  $t = t_f$  (bottom left). The evolution of the internal and external radius of the shell is also shown (bottom right) and compared with the analytical solution.

### 3.5. The Saltzman problem

Another classical test case for Lagrangian gas dynamics is the Saltzman problem, which was presented for the first time by Dukowicz et al. in [35] for a two-dimensional Cartesian grid that has been skewed in such a way that no element edges are aligned with the main fluid flow. It is a very challenging test problem against which any Lagrangian scheme ought to be validated [69, 64]. It involves a strong shock wave caused by a piston that is moving along the

Table 6: Absolute error for the internal and external radius location between exact  $R_{ex}$  and numerical  $R_{num}$  solution.

	$R_{ex}$	$R_{num}$	$ err $
<i>Internal radius</i>	0.45000000	0.44996063	3.94E-05
<i>External radius</i>	0.50000000	0.49930053	6.99E-04

main direction  $x$  of the initial rectangular domain  $\Omega(0) = [0; 1] \times [0; 0.1]$ , which is initially discretized by  $100 \times 10$  square elements. According to [69, 64], each element is then split into two *right triangles*, so that we obtain a total number of elements of  $N_E = 2 \cdot 100 \times 10 = 2000$ , and finally the following mapping is applied in order to skew the mesh:

$$x' = x + (0.1 - y) \sin(\pi x) \quad y' = y, \quad (65)$$

where  $\mathbf{x} = (x, y)$  represents the coordinate vector of the uniform grid, while  $\mathbf{x}' = (x', y')$  are the final skewed coordinates. As done in [64] the fluid is initially at rest and is assigned an internal energy  $e_0 = 10^{-4}$  and a density  $\rho_0 = 1$ , hence the initial condition in terms of conserved variables reads  $\mathbf{Q}_0 = (\rho_0, \rho u_0, \rho v_0, \rho E_0) = (1, 0, 0, 10^{-4})$ . According to [64], the ratio of specific heats is set to  $\gamma = \frac{5}{3}$  and the final time is assumed to be  $t_f = 0.6$ , while the piston is moving with velocity  $\mathbf{v}_p = (1, 0)$  towards the right boundary of the domain. Moving slip wall boundary condition is imposed on the piston, whereas fixed slip wall boundaries have been set on the remaining sides of the domain. As fully explained in [10, 87], the exact solution  $\mathbf{Q}_{ex}(\mathbf{x}, t)$  is computed by solving a one-dimensional Riemann problem and at the final time  $t_f$  it is given by

$$\mathbf{Q}_{ex}(\mathbf{x}, t_f) = \begin{cases} (4, 1, 0, 2.5) & \text{if } x \leq x_f, \\ (1, 0, 0, 10^{-4}) & \text{if } x > x_f, \end{cases} \quad (66)$$

where  $x_f = 0.8$  denotes the final shock location. The piston is moving very fast, so that the fluid next to the piston is highly compressed and elements there must typically obey a severe CFL condition. In practice, we have to start the simulation with CFL=0.1, hence using very small and *global* timesteps. After time  $t = 0.01$  the numerical scheme proceeds with the new time accurate *local time stepping* algorithm described in this article. We have used the third order version of our LTS Lagrangian ADER-WENO schemes and the very robust Rusanov-type numerical flux (46). Figure 12 shows a comparison between the exact and the numerical solution for density and horizontal velocity at the final time of the simulation for both the LTS and the GTS version of our algorithm, while the initial and the final mesh configurations are depicted in Figure 13. An overall good agreement of the numerical solution with the exact solution can be observed and the decrease of the density which occurs near the piston is due to the well known *wall-heating problem*, see [86]. Furthermore we point out that the results obtained with the LTS scheme given in the left column of Figure 12 do not differ very much from the numerical solution obtained with global time stepping (GTS) shown in the right column of Figure 12.

#### 4. Conclusions

In this article we have presented a high order Lagrangian finite volume schemes with time-accurate local time stepping (LTS) on moving unstructured triangular meshes. The numerical scheme is derived from [10, 38], where a classical global time stepping approach is adopted, and from the recently developed one-dimensional high order Lagrangian LTS numerical scheme [36]. In our approach the WENO reconstruction technique is used to achieve high order of accuracy in space, while high order of accuracy in time is obtained via the local space-time Galerkin predictor. The new algorithm illustrated in this article is based on a non-conforming mesh in time, with hanging nodes that are continuously moving and in principle never match the same time level, unless either an intermediate output time or the final time of the simulation is reached. As a consequence, the reconstruction is carried out locally, i.e. within each control volume, using a virtual geometry and a virtual set of cell averages of the surrounding elements that are both computed using the high order space-time predictor solution. In order to develop a fully conservative numerical scheme, the fluxes are evaluated relying on memory variables, which allow to record all fluxes accumulated



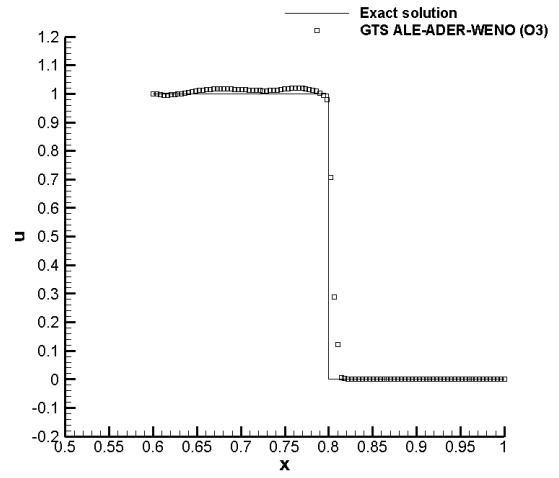
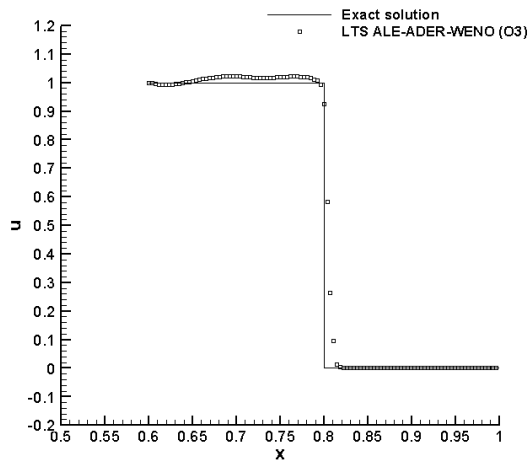
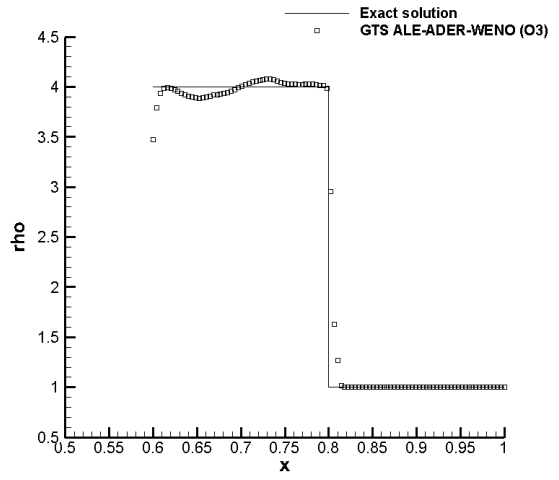
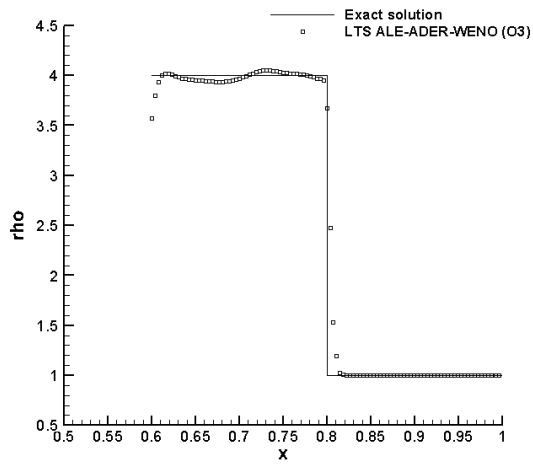


Figure 12: Third order accurate numerical solution for the Saltzman problem at the final time  $t_f = 0.6$ . *Left panels:* solution obtained with LTS. *Right panels:* solution obtained with GTS.

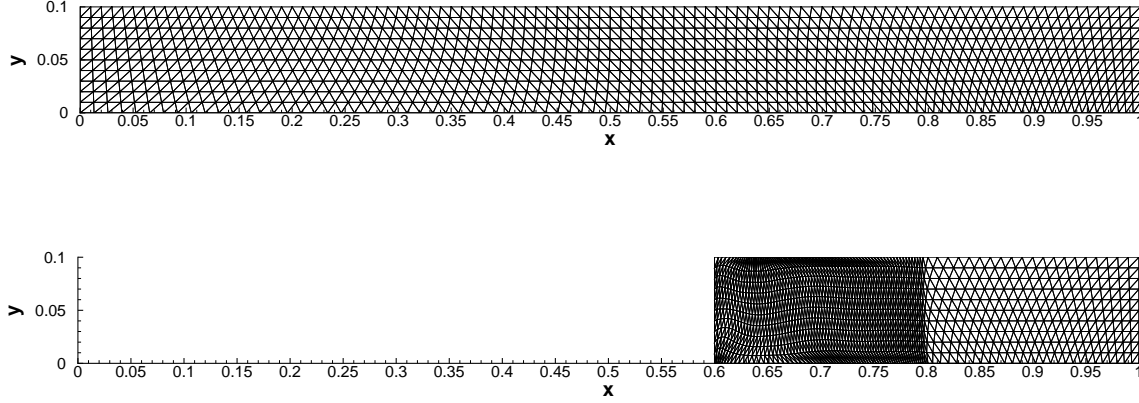


Figure 13: Initial and final mesh configuration for the Saltzman problem using the new LTS algorithm.

in the past within each control volume. Unlike the one-dimensional version of the algorithm presented in [36], on two-dimensional unstructured meshes we need also to compute additional fluxes over triangular space-time sub-surfaces, whenever an element performs the update timestep. This additional computational and algorithmic complexity is due to the increased complexity of the topology of a 2D mesh, which consists in control volumes, edges and nodes. By construction, our scheme is conservative and automatically satisfies the geometric conservation law (GCL) due to the integration over a closed space-time control volume.

The algorithm has been applied to the Euler equations of compressible gas dynamics, solving a set of canonical test problems and benchmarks for Lagrangian schemes. Furthermore convergence rates up to fourth order of accuracy in space and in time have been shown.

Further work may contain the extension of the presented LTS algorithm to three space dimensions and non-conservative hyperbolic balance laws as well as the implementation of a proper treatment for stiff source terms, hence allowing the scheme to be applied to more complex systems of equations.

## Acknowledgments

The presented research has been financed by the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP7/2007-2013) with the research project *STiMulUs*, ERC Grant agreement no. 278267. The authors acknowledge PRACE for awarding us access to the SuperMUC supercomputer of the Leibniz Rechenzentrum (LRZ) in Munich, Germany.

## References

- [1] T. Aboiyar, E.H. Georgoulis, and A. Iske. Adaptive ADER Methods Using Kernel-Based Polyharmonic Spline WENO Reconstruction. *SIAM Journal on Scientific Computing*, 32:3251–3277, 2010.
- [2] D. Avesani, M. Dumbser, and A. Bellin. A New Class of Moving-Least-Squares WENO SPH Schemes. *Journal of Computational Physics*, 270:278–299, 2014.
- [3] A. Baeza, A. Martínez-Gavara, and P. Mulet. Adaptation based on interpolation errors for high order mesh refinement methods applied to conservation laws. *Applied Numerical Mathematics*, 62:278–296, 2012.
- [4] A. Baeza and P. Mulet. Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations. *International Journal for Numerical Methods in Fluids*, 52:455–471, 2006.
- [5] D. Balsara and C.W. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *Journal of Computational Physics*, 160:405–452, 2000.
- [6] T.J. Barth and P.O. Frederickson. Higher order solution of the euler equations on unstructured grids using quadratic reconstruction. *28th Aerospace Sciences Meeting*, pages AIAA paper no. 90-0013, January 1990.

- [7] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64–84, May 1989.
- [8] M. J. Berger and J. Olinger. Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations. *Journal of Computational Physics*, 53:484, March 1984.
- [9] W. Boscheri, D.S. Balsara, and M. Dumbser. Lagrangian ADER-WENO Finite Volume Schemes on Unstructured Triangular Meshes Based On Genuinely Multidimensional HLL Riemann Solvers. *Journal of Computational Physics*, 267:112 – 138, 2014.
- [10] W. Boscheri and M. Dumbser. Arbitrary–Lagrangian–Eulerian One–Step WENO Finite Volume Schemes on Unstructured Triangular Meshes. *Communications in Computational Physics*, 14:1174–1206, 2013.
- [11] W. Boscheri and M. Dumbser. Lagrangian ADER-WENO Finite Volume Schemes on Unstructured Tetrahedral Meshes for Conservative and Nonconservative Hyperbolic Systems in 3D. *Journal of Computational Physics*, 275:484–523, 2014.
- [12] W. Boscheri, M. Dumbser, and D.S. Balsara. High Order Lagrangian ADER-WENO Schemes on Unstructured Meshes – Application of Several Node Solvers to Hydrodynamics and Magnetohydrodynamics. *International Journal for Numerical Methods in Fluids*. in press. DOI: 10.1002/ld.3947.
- [13] J. Breil, T. Haribey, P.H. Maire, and M.J. Shashkov. A multi-material ReALE method with MOF interface reconstruction. *Computers and Fluids*, 83:115–125, 2013.
- [14] R. Bürger, P. Mulet, and L.M. Villada. Spectral weno schemes with adaptive mesh refinement for models of polydisperse sedimentation. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für angewandte Mathematik und Mechanik*, pages n/a–n/a, 2012.
- [15] G. Carré, S. Del Pino, B. Després, and E. Labourasse. A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics*, 228:5160–5183, 2009.
- [16] C.E. Castro, M. Käser, and E.F. Toro. Space–time adaptive numerical methods for geophysical applications. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367:4613–4631, 2009.
- [17] V. Casulli. A semi-implicit finite difference method for non-hydrostatic free-surface flows. *International Journal for Numerical Methods in Fluids*, 30:425–440, 1999.
- [18] V. Casulli. A semi-implicit numerical method for the free–surface Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 74:605–622, 2014.
- [19] V. Casulli and G. S. Stelling. Semi-implicit subgrid modelling of three-dimensional free-surface flows. *International Journal for Numerical Methods in Fluids*, 67:441–449, 2011.
- [20] J. Cesenek, M. Feistauer, J. Horacek, V. Kucera, and J. Prokopova. Simulation of compressible viscous flow in time-dependent domains. *Applied Mathematics and Computation*, 219:7139–7150, 2013.
- [21] J. Cheng and C.W. Shu. A high order ENO conservative Lagrangian type scheme for the compressible Euler equations. *Journal of Computational Physics*, 227:1567–1596, 2007.
- [22] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for systems of conservation laws multi-dimensional optimal order detection (mood). *Journal of Computational Physics*, 230:4028 – 4050, 2011.
- [23] A. Claisse, B. Després, E. Labourasse, and F. Ledoux. A new exceptional points method with application to cell-centered Lagrangian schemes and curved meshes. *Journal of Computational Physics*, 231:4324–4354, 2012.
- [24] B. Cockburn, G. E. Karniadakis, and C.W. Shu. *Discontinuous Galerkin Methods*. Lecture Notes in Computational Science and Engineering. Springer, 2000.
- [25] B. Després and C. Mazeran. Symmetrization of Lagrangian gas dynamic in dimension two and multidimensional solvers. *C.R. Mecanique*, 331:475–480, 2003.
- [26] B. Després and C. Mazeran. Lagrangian gas dynamics in two-dimensions and Lagrangian systems. *Archive for Rational Mechanics and Analysis*, 178:327–372, 2005.
- [27] S. Diot, S. Clain, and R. Loubère. Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials. *Computers and Fluids*, 64:43–63, 2012.
- [28] V.A. Dobrev, T.E. Ellis, Tz.V. Kolev, and R.N. Rieben. Curvilinear Finite elements for Lagrangian hydrodynamics. *International Journal for Numerical Methods in Fluids*, 65:1295–1310, 2011.
- [29] V.A. Dobrev, T.E. Ellis, Tz.V. Kolev, and R.N. Rieben. High Order Curvilinear Finite Elements for Lagrangian Hydrodynamics. *SIAM Journal on Scientific Computing*, 34:606–641, 2012.
- [30] V.A. Dobrev, T.E. Ellis, Tz.V. Kolev, and R.N. Rieben. High Order Curvilinear Finite Elements for axisymmetric Lagrangian Hydrodynamics. *Computers and Fluids*, 83:58–69, 2013.
- [31] V. Dolejsi. Semi-implicit interior penalty discontinuous Galerkin methods for viscous compressible flows. *Communications in Computational Physics*, 4:231–274, 2008.
- [32] V. Dolejsi and M. Feistauer. A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow. *Journal of Computational Physics*, 198:727–746, 2004.
- [33] V. Dolejsi, M. Feistauer, and J. Hozman. Analysis of semi-implicit DGFEM for nonlinear convection-diffusion problems on nonconforming meshes. *Computer Methods in Applied Mechanics and Engineering*, 196:2813–2827, 2007.
- [34] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6:345–390, 1991.
- [35] J.K. Dukovicz and B. Meltz. Vorticity errors in multidimensional lagrangian codes. *Journal of Computational Physics*, 99:115 – 134, 1992.
- [36] M. Dumbser. Arbitrary-Lagrangian-Eulerian ADER-WENO Finite Volume Schemes with Time-Accurate Local Time Stepping for Hyperbolic Conservation Laws. *Computational Methods in Applied Mechanics and Engineering*, 280:57–83, 2014.
- [37] M. Dumbser, D.S. Balsara, E.F. Toro, and C.-D. Munz. A unified framework for the construction of one-step finite volume and discontinuous galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227:8209 – 8253, 2008.
- [38] M. Dumbser and W. Boscheri. High-order unstructured Lagrangian one-step WENO finite volume schemes for non-conservative hyperbolic systems: Applications to compressible multi-phase flows. *Computers and Fluids*, 86:405 – 432, 2013.
- [39] M. Dumbser, A. Hidalgo, and O. Zanotti. High order space-time adaptive ADER-WENO finite volume schemes for non-conservative hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 268:359–387, 2014.

- [40] M. Dumbser and M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221:693–723, 2007.
- [41] M. Dumbser, M. Käser, V.A. Titarev, and E.F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226:204–243, 2007.
- [42] M. Dumbser, M. Käser, and E. F. Toro. An arbitrary high order discontinuous Galerkin method for elastic waves on unstructured meshes V: Local time stepping and  $p$ -adaptivity. *Geophysical Journal International*, 171:695–717, 2007.
- [43] M. Dumbser and E. F. Toro. On universal Osher–type schemes for general nonlinear hyperbolic conservation laws. *Communications in Computational Physics*, 10:635–671, 2011.
- [44] M. Dumbser, A. Uuriintsetseg, and O. Zanotti. On Arbitrary–Lagrangian–Eulerian One–Step WENO Schemes for Stiff Hyperbolic Balance Laws. *Communications in Computational Physics*, 14:301–327, 2013.
- [45] M. Dumbser, O. Zanotti, A. Hidalgo, and D. S. Balsara. ADER–WENO finite volume schemes with space-time adaptive mesh refinement. *Journal of Computational Physics*, 248:257–286, 2013.
- [46] M. Feistauer, J. Horacek, M. Ruzicka, and P. Svacek. Numerical analysis of flow-induced nonlinear vibrations of an airfoil with three degrees of freedom. *Computers and Fluids*, 49:110–127, 2011.
- [47] A. Ferrari, C.D. Munz, and B. Weigand. A high order sharp interface method with local timestepping for compressible multiphase flows. *Communications in Computational Physics*, 9:205–230, 2011.
- [48] J. Flaherty, R. Loy, M. Shephard, B. Szymanski, J. Teresco, and L. Ziantz. Adaptive local refinement with octree load–balancing for the parallel solution of three–dimensional conservation laws. *Journal of Parallel and Distributed Computing*, 47:139–152, 1997.
- [49] M.M. Francois, M.J. Shashkov, T.O. Masser, and E.D. Dendy. A comparative study of multimaterial Lagrangian and Eulerian methods with pressure relaxation. *Computers and Fluids*, 83:126–136, 2013.
- [50] F.Vilar. Cell-centered discontinuous Galerkin discretization for two-dimensional Lagrangian hydrodynamics. *Computers and Fluids*, 64:64–73, 2012.
- [51] F.Vilar, P.H. Maire, and R. Abgrall. Cell-centered discontinuous Galerkin discretizations for two-dimensional scalar conservation laws on unstructured grids and for one-dimensional Lagrangian hydrodynamics. *Computers and Fluids*, 46(1):498–604, 2010.
- [52] F.Vilar, P.H. Maire, and R. Abgrall. A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids. *Journal of Computational Physics*, 276:188–234, 2014.
- [53] G. Gassner, F. Lörcher, and C. D. Munz. A discontinuous Galerkin scheme based on a space-time expansion II. viscous flow equations in multi dimensions. *Journal of Scientific Computing*, 34:260–286, 2008.
- [54] M.J. Grote and T. Mitkova. Explicit local time-stepping methods for Maxwell’s equations. *Journal of Computational and Applied Mathematics*, 234:3283–3302, 2010.
- [55] M.J. Grote and T. Mitkova. High-order explicit local time-stepping methods for damped wave equations. *Journal of Computational and Applied Mathematics*, 239:270–289, 2013.
- [56] A. Hidalgo and M. Dumbser. ADER schemes for nonlinear systems of stiff advectiondiffusionreaction equations. *Journal of Scientific Computing*, 48:173–189, 2011.
- [57] C. Hu and C.W. Shu. A high-order weno finite difference scheme for the equations of ideal magnetohydrodynamics. *Journal of Computational Physics*, 150:561 – 594, 1999.
- [58] G.S. Jiang and C.W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202–228, 1996.
- [59] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp Element Methods in CFD*. Oxford University Press, 1999.
- [60] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205:486 – 508, 2005.
- [61] R.E. Kidder. Laser-driven compression of hollow shells: power requirements and stability limitations. *Nucl. Fus.*, 1:3 – 14, 1976.
- [62] L. Krivodonova. An efficient local time–stepping scheme for solution of nonlinear conservation laws. *Journal of Computational Physics*, 229:8537–8551, 2010.
- [63] Z. Li, X. Yu, and Z. Jia. The cell–centered discontinuous Galerkin method for Lagrangian compressible Euler equations in two dimensions. *Computers and Fluids*, 96:152–164, 2014.
- [64] W. Liu, J. Cheng, and C.W. Shu. High order conservative Lagrangian schemes with LaxWendroff type time discretization for the compressible Euler equations. *Journal of Computational Physics*, 228:8872–8891, 2009.
- [65] F. Lörcher, G. Gassner, and C. D. Munz. A discontinuous Galerkin scheme based on a space-time expansion. I. inviscid compressible flow in one space dimension. *Journal of Scientific Computing*, 32:175–199, 2007.
- [66] R. Loubère, M. Dumbser, and S. Diot. A new family of high order unstructured mood and ader finite volume schemes for multidimensional systems of hyperbolic conservation laws. *Communications in Computational Physics*, 16:718–763, 2014.
- [67] R. Loubère, P.H. Maire, and P. Váchal. A second–order compatible staggered Lagrangian hydrodynamics scheme using a cell–centered multidimensional approximate Riemann solver. *Procedia Computer Science*, 1:1931–1939, 2010.
- [68] P.H. Maire. A high-order cell–centered Lagrangian scheme for compressible fluid flows in two–dimensional cylindrical geometry . *Journal of Computational Physics*, 228:6882–6915, 2009.
- [69] P.H. Maire. A high-order cell-centered lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *Journal of Computational Physics*, 228:2391–2425, 2009.
- [70] P.H. Maire. A high-order one-step sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids. *Computers and Fluids*, 46(1):341–347, 2011.
- [71] P.H. Maire. A unified sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids. *International Journal for Numerical Methods in Fluids*, 65:1281–1294, 2011.
- [72] P.H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered lagrangian scheme for two-dimensional compressible flow problems. *SIAM Journal on Scientific Computing*, 29:1781–1824, 2007.
- [73] P.H. Maire and J. Breil. A second-order cell-centered lagrangian scheme for two-dimensional compressible flow problems. *International Journal for Numerical Methods in Fluids*, 56:1417–1423, 2007.

- [74] P.H. Maire and B. Nkonga. Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics. *Journal of Computational Physics*, 228:799–821, 2009.
- [75] C.D. Munz. On Godunov–type schemes for Lagrangian gas dynamics. *SIAM Journal on Numerical Analysis*, 31:17–42, 1994.
- [76] C. Olliver-Gooch and M. Van Altena. A high-order–accurate unstructured mesh finite–volume scheme for the advection–diffusion equation. *Journal of Computational Physics*, 181:729 – 752, 2002.
- [77] A. López Ortega and G. Scovazzi. A geometrically–conservative, synchronized, flux–corrected remap for arbitrary Lagrangian–Eulerian computations with nodal finite elements. *Journal of Computational Physics*, 230:6709–6741, 2011.
- [78] S.K. Sambasivan, M.J. Shashkov, and D.E. Burton. A finite volume cell-centered Lagrangian hydrodynamics approach for solids in general unstructured grids. *International Journal for Numerical Methods in Fluids*, 72:770–810, 2013.
- [79] S.K. Sambasivan, M.J. Shashkov, and D.E. Burton. Exploration of new limiter schemes for stress tensors in Lagrangian and ALE hydrocodes. *Computers and Fluids*, 83:98–114, 2013.
- [80] G. Scovazzi. Lagrangian shock hydrodynamics on tetrahedral meshes: A stable and accurate variational multiscale approach. *Journal of Computational Physics*, 231:8029–8069, 2012.
- [81] A.H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.
- [82] A. Taube, M. Dumbser, C.D. Munz, and R. Schneider. A High Order Discontinuous Galerkin Method with Local Time Stepping for the Maxwell Equations. *International Journal Of Numerical Modelling: Electronic Networks, Devices And Fields*, 22:77–103, 2009.
- [83] V.A. Titarev and E.F. Toro. ADER schemes for three-dimensional nonlinear hyperbolic systems. *Journal of Computational Physics*, 204:715–736, 2005.
- [84] V.A. Titarev, P. Tsoutsanis, and D. Drikakis. WENO schemes for mixed–element unstructured meshes. *Communications in Computational Physics*, 8:585–609, 2010.
- [85] E. F. Toro and V. A. Titarev. Derivative Riemann solvers for systems of conservation laws and ADER methods. *Journal of Computational Physics*, 212(1):150–165, 2006.
- [86] E.F. Toro. Anomalies of conservative methods: analysis, numerical evidence and possible cures. *International Journal of Computational Fluid Dynamics*, 11:128–143, 2002.
- [87] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction*. Springer, 2009.
- [88] P. Tsoutsanis, V.A. Titarev, and D. Drikakis. WENO schemes on arbitrary mixed-element unstructured meshes in three space dimensions. *Journal of Computational Physics*, 230:1585–1601, 2011.
- [89] J. Uitzmann, T. Schwartzkopff, M. Dumbser, and C.D. Munz. Heterogeneous Domain Decomposition for Computational Aeroacoustics. *AIAA Journal*, 44:2231–2250, 2006.
- [90] Y.V. Yanilkin, E.A. Goncharov, V.Y. Kolobyanin, V.V. Sadchikov, J.R. Kamm, M.J. Shashkov, and W.J. Rider. Multi-material pressure relaxation methods for lagrangian hydrodynamics. *Computers and Fluids*, 83:137–143, 2013.
- [91] O. Zanotti and M. Dumbser. A high order special relativistic hydrodynamic code with space-time adaptive mesh refinement. *ArXiv e-prints*, December 2013.
- [92] Y.T. Zhang and C.W. Shu. Third order WENO scheme on three dimensional tetrahedral meshes. *Communications in Computational Physics*, 5:836–848, 2009.